



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

**IMPLEMENTASI DAN PENGUJIAN SISTEM
MONITORING KESEHATAN STRUKTUR PADA
JEMBATAN SURABAYA-MADURA**

Arvid Prasetya Cahya
NRP 2210 100 003

Dosen Pembimbing
Eko Setijadi, ST. MT. Ph.D.
Ir. Gatot Kusrahardjo, MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 141599

**IMPLEMENTATION AND TESTING OF THE
SYSTEM HEALTH MONITORING OF
STRUCTURES ON THE SURABAYA-
MADURA BRIDGE**

Arvid Prasetya Cahya
NRP 2210 100 003

Supervisors
Eko Setijadi, ST. MT. Ph.D.
Ir. Gatot Kusrahardjo, MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

**IMPLEMENTASI DAN PENGUJIAN SISTEM
MONITORING KESEHATAN STRUKTUR PADA
JEMBATAN SURABAYA-MADURA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Telekomunikasi Multimedia
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

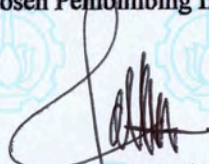
Menyetujui

Dosen Pembimbing I,



Eko Setijadi, ST., MT., Ph.D.
NIP. 197210012003121002

Dosen Pembimbing II,



Ir. Gatot Kusrahardjo, MT.
NIP. 195904281986011001



IMPLEMENTASI DAN PENGUJIAN SISTEM MONITORING KESEHATAN STRUKTUR PADA JEMBATAN SURABAYA-MADURA

Arvid Prasetya Cahya
2210 100 003

Dosen Pembimbing I : Eko Setijadi, ST. MT. Ph.D.
Dosen Pembimbing II : Ir. Gatot Kusrahardjo, MT.

ABSTRAK

Jembatan merupakan salah satu bangunan infrastruktur yang sangat penting. Oleh karena itu jembatan memiliki banyak manfaat mulai dari pemerataan pembangunan, meningkatkan taraf sosial, hingga meningkatkan aktivitas perekonomian suatu daerah. SHMS (Structure Health Monitoring System) merupakan salah satu sistem yang digunakan untuk memonitor dan mendeteksi kerusakan pada suatu bangunan dan infrastruktur secara real time.

Pada tugas akhir ini *wireless sensor network* digunakan untuk sistem komunikasi data pada jembatan bentang panjang Surabaya-Madura untuk kemudian dapat diakses pada pusat pengawasan secara nirkabel. Sistem ini terdiri dari dua skenario, pada skenario pertama menggunakan satu node sensor (*end device*), pada skenario kedua menggunakan dua node sensor (*end device*) dan satu node koordinator (*coordinator*). Node sensor berfungsi sebagai pengolah data pembacaan sensor dan mengirimkan pada node koordinator. Sedangkan node koordinator berfungsi sebagai penerima data pembacaan sensor dari node sensor untuk kemudian dikirimkan pada server dan disimpan pada database sehingga data dapat diakses secara nirkabel melalui jaringan WLAN (*Wireless Local Area Network*).

Unjuk kerja dari perangkat yang telah diintegrasikan menunjukkan *throughput* node sensor 121.77 Bps, *throughput* node sensor1 48.048 Bps, *throughput* node sensor2 39.534 Bps, *packet loss* sensor 38.486%, *packet loss* sensor1 75.732%, *packet loss* sensor2 80.03% dan *delay* sensor 1.26 detik, *delay* sensor1 1.2 detik, *delay* sensor2 1.02 detik.

Kata Kunci : Jaringan sensor nirkabel, SHMS, *end device*, *coordinator*

IMPLEMENTATION AND TESTING OF THE SYSTEM HEALTH MONITORING OF STRUCTURES ON THE SURABAYA- MADURA BRIDGE

Arvid Prasetya Cahya
2210 100 003

Supervisor I : Eko Setijadi, ST. MT. Ph.D.
Supervisor II : Ir. Gatot Kusrahardjo, MT.

ABSTRACT

The bridge is one of the building infrastructure is very important. Therefore the bridge has many benefits ranging from equitable development, improve social, to improve the economy of an area. SHMS (Structure Health Monitoring System) is a system used to monitor and detect damage to buildings and infrastructure in real time.

In this final task wireless sensor network used for data communication system on longspan bridge of Surabaya-Madura to can than be accessed wirelessly surveillance center. This system consist of two scenarios, in the first scenario using a single sensor node (end device), in the second scenario uses two sensor nodes (end device) and one node coordinator. Node data processing functions as a sensor reading of sensors and transmit on the node coordinator. While node coordinator serves as the sensor readings from the data receiver node sensors for late sent on a dedicated server and stored in the database so that the data can be accessed wirelessly through a network of WLAN (Wireless Local Area Network).

Performance of the device has been integrated sensor nodes throughput shows 121.77 Bps, throughput sensor1 48.048 Bps, throughput sensor2 39.534 Bps, packet loss sensor 38.486%, packet loss sensor1 75.732%, Packet loss sensor2 80.03%, and delay sensor 1.26 sec, delay sensor1 1.2 sec, delay sensor2 1.02 sec.

Keyword: *Wireless Sensor Network, SHMS, end device, coordinator*

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul **“IMPLEMENTASI DAN PENGUJIAN SISTEM MONITORING KESEHATAN STRUKTUR PADA JEMBRAN SURABAYA-MADURA”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 2016

Arvid Prasetya Cahya
NRP. 2210100003

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah SWT serta tak lupa shalawat serta salam dihaturkan kepada Nabi Muhammad SAW, keluarga, sahabat dan pengikut beliau hingga akhir zaman. Karena atas rahmat dan karunia – Nya penulis dapat menyelesaikan penulisan buku Tugas Akhir dengan judul :


“IMPLEMENTASI DAN PENGUJIAN SISTEM MONITORING KESEHATAN STRUKTUR PADA JEMBATAN SURABAYA-MADURA”

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan dalam menyelesaikan pendidikan Strata – 1 pada Bidang Studi Telekomunikasi Multimedia, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh nopember Surabaya.

Penulis menyadari bahwa dalam penulisan Tugas Akhir ini banyak mengalami kendala, namun berkat bantuan, bimbingan dan kerjasama dari berbagai pihak segala kendala dapat teratasi. Untuk itu penulis menyampaikan banyak terima kasih kepada :

1. Kedua orang tua, Mirzah Heru Cahya dan Sri Rahayu Setyowati
2. Bapak Eko Setijadi dan Bapak Gatot Kusrahardjo selaku Dosen Pembimbing atas segala bantuan, perhatian, dan arahan selama pengerjaan Tugas Akhir ini.
3. Bapak Endroyono selaku Koordinator Bidang Studi Telekomunikasi Multimedia Jurusan Teknik Elektro ITS.
4. Bapak Ardyono Priyadi selaku Ketua Jurusan Teknik Elektro ITS.
5. Bapak dan Ibu dosen bidang studi Telekomunikasi Multimedia, Teknik Elektro ITS
6. Rekan-rekan e50 dan seluruh elemen warga Teknik Elektro ITS atas segala motivasi yang selalu diberika.

Penulis menyadari bahwa pada penyusunan laporan tugas akhir ini masih terdapat kekurangan-kekurangan karena keterbatasan kemampuan yang penulis miliki, walaupun demikian penulis berharap tugas akhir ini dapat bermanfaat bagi yang membutuhkannya.



Surabaya, 2015

Penulis

Table of Contents

Contents

TITLE	i
LEGAL APPROVEMENT.....	v
APPROVAL SHEET	vii
ABSTRACT in BAHASA	ix
ABSTRACT in ENGLISH.....	xi
PREFACE.....	xiii
TABLE of CONTENT	xv
ILLUSTRATION LIST	xviii
TABLE LIST	xx
CHAPTER 1	1
1.1 Background	1
1.2 Problems	2
1.3 Problem Limitation	2
1.4 Purpose	2
1.5 Methodology	3
1.6 Writing's Systemathics.....	4
1.7 Relevance.....	4
CHAPTER 2	5
2.1 Structural Health Monitoring System (SHMS).....	5
2.2 Surabaya-Madura Bridge	6
2.3 Wireless Sensor Network (WSN).....	9
2.3.1 <i>Wireless Sensor Network Single-hop</i>	11
2.4 Wireless Communication Standarts	12
2.4.1 Standar IEEE 802.15.4 (<i>ZigBee</i>)	13
2.4.2 Standar IEEE 802.11	14
2.5 Accelerometer	16
2.6 Arduino	18
2.6.1 Arduino Uno	18
2.6.2 Arduino Due.....	18
2.6.3 Arduino Ethernet Shield.....	19
2.6.4 <i>Arduino Xbee Shield</i>	19
2.7 <i>Xbee</i>	20
2.8 NanoStation M2	21
2.9 XCTU	22
2.10 Webserver.....	23
2.10.1 Webserver Apache.....	23
2.10.2 Database SQL.....	24

2.10.3	PHP Programming	24
2.11	Network Performance Parameter	25
2.11.1	<i>Packet Loss</i>	25
2.11.2	<i>Troughput</i>	25
2.11.3	<i>Delay</i>	26
CHAPTER 3		27
3.1	<i>Wireless Sensor Network application for bridge</i>	28
3.2	Hardware Modelling.....	31
3.2.1	Sensor Node Modelling.....	31
3.2.2	Coordinator Node Modelling	33
3.3	Haardware Integration.....	34
3.3.1	Sensor Node Hardware Integration	35
3.3.2	Coordinator Node Hardware Integration	36
3.4	Web and Database Modelling.....	37
3.4.1	Webserver	37
3.4.2	PHP	41
3.5	Communication Network Modelling	
	<i>Wireless Local Area Network</i>	41
3.6	System Performance	44
CHAPTER 4		47
4.1	Sensor Node Test (<i>end device</i>).....	47
4.2	Coordinator Node Test	49
4.3	NanoStation M2 Test	50
4.4	Sensor Node Communication Test	52
4.4.1	Sensor Node Test on 1 st Scenario	52
4.4.2	Sensor Node Test on 2 nd Scenario	57
4.5	Sensor Node Delay Test	65
4.6	System Analisyst	66
CHAPTER 5		69
5.1	Conclusion	69
5.2	Advice	70
BIBLIOGRAPHY		71
WRITER'S BIOGRAPHY		73
APPENDIX		75
	Listing Program Mikrokontroler Arduino as Sensor Node	75
	Listing Program Mikrokontroler Arduino as Coordinator Node....	78
	Listing Program Python.....	78
	Listing Program PHP.....	79

DAFTAR GAMBAR

Gambar 2.1 <i>Structural Health Monitoring System</i> pada jembatan.....	5
Gambar 2.2 SHMS Jembatan Suramadu.....	7
Gambar 2.3 Penempatan Sensor SHMS Jembatan Suramadu[1].....	8
Gambar 2.4 <i>Backbone</i> jaringan SHMS Jembatan Suramadu[1].....	8
Gambar 2.5 Topologi jaringan pada SHMS Jembatan Suramadu.....	9
Gambar 2.6 Topologi jaringan <i>Star</i>	10
Gambar 2.7 Topologi jaringan <i>Tree</i>	11
Gambar 2.8 Topologi jaringan <i>Mesh</i>	11
Gambar 2.9 Jaringan <i>multi-hop</i>	12
Gambar 2.10 Jaringan <i>single-hop</i>	12
Gambar 2.11 Arsitektur Protokol ZigBee.....	13
Gambar 2.12 Arsitektur Protokol WLAN.....	16
Gambar 2.13 Accelerometer.....	17
Gambar 2.14 Arduino Uno.....	18
Gambar 2.15 Arduino Due.....	19
Gambar 2.16 Arduino Ethernet Shield.....	19
Gambar 2.17 Arduino Xbee Shield.....	20
Gambar 2.18 <i>Xbee</i>	21
Gambar 2.19 NanoStation M2.....	21
Gambar 2.20 Tampilan <i>XCTU</i>	22
Gambar 3.1 Diagram alir pengerjaan tugas akhir (a) start hingga integrasi penyusun <i>node</i> (b) integrasi antar <i>node</i>	28
Gambar 3.2 Blok diagram komunikasi data WSN.....	29
Gambar 3.3 Arsitektur sistem komunikasi pada SHMS jembatan.....	30
Gambar 3.4 Diagram alir node sensor (<i>end device</i>).....	31
Gambar 3.5 Sensor accelerometer MMA 7361.....	32
Gambar 3.6 Arduino Uno dan Due.....	32
Gambar 3.7 <i>Xbee</i>	33
Gambar 3.8 Diagram alir node koordinator.....	34
Gambar 3.9 Perangkat node sensor.....	35
Gambar 3.10 Perangkat node koordinator.....	36
Gambar 3.11 Xampp control panel.....	37
Gambar 3.12 Halaman utama Xampp.....	38
Gambar 3.13 Halaman utama phpmyadmin.....	38
Gambar 3.14 Tampilan perangkat lunak python (GUI).....	39
Gambar 3.15 Tampilan database MySQL.....	40
Gambar 3.16 Tampilan PHP.....	41
Gambar 3.17 Halaman nanostation2.....	42

Gambar 3.18 Halaman Pengaturan nanostation m2	43
Gambar 4.1 pengujian pada (a) skenario 1, (b) skenario 2.....	47
Gambar 4.2 Tampilan pemrograman arduino	48
Gambar 4.3 <i>window serial monitoring</i> pada node sensor	49
Gambar 4.4 window serial monitoring pada node coordinator.....	50
Gambar 4.5 nanostation sebagai access point (a), nanostation sebagai station (b).....	52
Gambar 4.6 Pengujian komunikasi pada skenario ke-1	53
Gambar 4.7 Grafik pengukuran throughput node sensor	55
Gambar 4.8 Grafik pengukuran packet loss node sensor	56
Gambar 4.9 Pengujian komunikasi pada skenario ke-2	57
Gambar 4.10 Grafik pengukuran throughput node sensor 1.....	60
Gambar 4.11 Grafik pengukuran packet loss node sensor 1.....	60
Gambar 4.12 Grafik pengukuran throughput node sensor 2.....	63
Gambar 4.13 Grafik pengukuran packet loss node sensor 2.....	64
Gambar 4.14 Tampilan database MySQL	65
Gambar 4.15 Data error pada database.....	67

DAFTAR TABEL

Tabel 2.1 Jenis standar WLAN	15
Tabel 3.1 Pengukuran rata-rata daya terima access point	44
Tabel 4.1 Pengukuran node sensor pada skenario ke-1	53
Tabel 4.2 Pengukuran rata-rata throughput dan packet loss node sensor	56
Tabel 4.3 Pengukuran node sensor 1 pada skenario ke-2	58
Tabel 4.4 Pengukuran rata-rata throughput dan packet loss node sensor1	61
Tabel 4.5 Pengukuran node sensor 2 pada skenario ke-2	61
Tabel 4.6 Pengukuran rata-rata throughput dan packet loss node sensor2	64
Tabel 4.7 Pengukuran rata-rata delay sensor tiap pengukuran.....	66

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jembatan merupakan salah satu bangunan infrastruktur yang sangat penting. Jembatan sebagai penghubung antara dua daerah yang saling terpisah. Oleh karena itu jembatan memiliki banyak manfaat mulai dari pemerataan pembangunan, meningkatkan taraf sosial, hingga meningkatkan aktivitas perekonomian suatu daerah.

Jembatan menurut bentangnya dapat dikasifikasi menjadi tiga yaitu jembatan bentang pendek ($<40\text{m}$), jembatan bentang menengah ($40\text{m}-125\text{m}$) dan jembatan bentang panjang ($>125\text{m}$).

Proses pengawasan pada suatu bangunan infrastruktur seperti jembatan menjadi sangat penting untuk mengetahui kondisi struktur jembatan dan mencegah kerusakan secara dini. Karena penurunan kemampuan dan struktur jembatan pasti terjadi karena berbagai faktor yang terjadi pada jembatan. Saat ini beberapa jembatan bentang panjang didunia menerapkan SHMS (*Structural Health Monitoring System*) untuk mempermudah proses pengawasan. Sistem tersebut dianggap lebih cepat dan efisien daripada menggunakan proses pengawasan manual karena menggunakan sensor-sensor yang memungkinkan untuk melakukan pengambilan data secara efektif dan efisien.

Jembatan Surabaya-Madura yang masuk dalam kategori bentang panjang (memiliki panjang total jembatan 5.438m) telah menerapkan SHMS (*Structural Health Monitoring System*) dalam proses pengawasan. Berbagai jenis sensor terpasang pada jembatan untuk melakukan pengawasan secara *real-time*. Sensor-sensor tersebut terhubung pada suatu perangkat akuisisi data untuk kemudian dikirimkan pada database melalui media kabel sehingga data dari sensor tersebut dapat digunakan untuk menganalisa kemampuan struktur jembatan.

Kondisi tersebut memunculkan alternatif sistem komunikasi data yaitu menggunakan WSN. WSN (*Wireless Sensor Network*) adalah suatu jaringan yang terdiri dari beberapa sensor yang berada pada lokasi berbeda, dimana sensor-sensor tersebut melakukan proses pengawasan pada suatu obyek tertentu dan melakukan pengiriman data secara nirkabel.

Pada tugas akhir ini *wireless sensor network* digunakan untuk sistem komunikasi data pada sensor yang dipasang pada jembatan bentang

panjang Surabaya-Madura untuk kemudian dapat diakses pada pusat pengawasan secara nirkabel. Dengan menggunakan *wireless sensor network* maka proses komunikasi data antara sensor dan pusat monitoring dapat terjadi secara efektif dan efisien.

Pada tugas akhir ini menggunakan dua jenis standar komunikasi data. Komunikasi data yang pertama menggunakan sistem *wireless sensor network* dimana setiap node sensor akan mengirimkan data menuju *datalogger* menggunakan protokol IEEE 802.15.4. Selanjutnya *datalogger* akan terhubung dengan *access point* dan jaringan lokal menggunakan protokol IEEE 802.11.

1.2 Permasalahan

Bagaimana merancang sistem komunikasi data dan protokol pengiriman data menggunakan *wireless sensor network* pada Jembatan bentang panjang Surabaya-Madura sehingga data pada sensor dapat diakses secara nirkabel?

1.3 Batasan Masalah

Agar dihasilkan suatu sistem yang berjalan secara efektif, maka penulis membatasi masalah sebagai berikut :

1. Sensor yang digunakan pada tugas akhir ini adalah sensor accelerometer MMA7361 untuk mendapatkan data getaran dari jembatan.
2. Penelitian tugas akhir ini melingkupi komunikasi data antara node sensor dengan node koordinator menggunakan WSN dan field server dengan client menggunakan WLAN.
3. Komunikasi data pada node sensor dan node koordinator menggunakan protokol IEEE 802.15.4
4. Komunikasi data pada jaringan lokal menggunakan protokol IEEE 802.11

1.4 Tujuan

Penelitian pada tugas akhir ini bertujuan untuk merancang sistem komunikasi data dengan menggunakan *wireless sensor network* pada Jembatan bentang-panjang Surabaya Madura sehingga data pada sensor dapat diakses secara nirkabel.

1.5 Metodologi

Penelitian pada tugas akhir ini dilakukan dengan metode penelitian sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan studi literatur mengenai sistem *wireless sensor network* dan penggunaannya pada SHMS (*Structural Health Monitoring System*). Studi literatur dilakukan dengan menganalisa dan membandingkan beberapa hasil penelitian-penelitian yang telah dilakukan sebelumnya. Tujuan utama dari tahap ini adalah untuk mengetahui jenis perangkat keras dan perangkat lunak yang memungkinkan untuk diterapkan pada sistem.

2. Perancangan perangkat keras

Setelah melakukan analisa dan diskusi berdasarkan beberapa penelitian-penelitian yang telah dilakukan sebelumnya, dibuat rancangan perangkat keras yang digunakan pada penelitian tugas akhir ini, meliputi penentuan sensor dan mikrokontroler yang digunakan yaitu sensor accelerometer dan arduino due dengan arduino ethernet shield untuk komunikasi data menggunakan protokol IEEE 802.11 serta xbee untuk komunikasi data *ZigBee*.

3. Perancangan perangkat lunak

Setelah melakukan perancangan perangkat keras yang digunakan pada sistem, dilakukan perancangan *gateway*, perancangan *database* sistem serta perancangan *website* untuk menampilkan data yang didapat oleh perangkat keras.

4. Implementasi hasil perancangan

Tahap ini adalah pengimplementasian dari perancangan perangkat keras dan perangkat lunak yang dilakukan pada tahap sebelumnya. Dimana hasil rancangan tersebut diintegrasikan.

5. Pengujian Komunikasi Data

Setelah dilakukan pengintegrasian perangkat, dilakukan pengujian untuk mengetahui performa dari sistem komunikasi data *wireless sensor network* apakah sesuai dengan parameter yang telah ditentukan.

6. Analisa data

Pada tahap ini dilakukan pengamatan dan analisa data terhadap data yang telah diperoleh setelah proses pengujian untuk kemudian diambil kesimpulan pada tahapan berikutnya.

7. Penarikan Kesimpulan

Tahap ini adalah tahap akhir dari penelitian dimana setelah dilakukan pengujian sistem didapatkan hasil yang akan direkomendasikan berupa konfigurasi jaringan pada sistem komunikasi data untuk modul *wireless sensor network* pada jembatan bentang panjang Surabaya-Madura.

1.6 Sistematika Penulisan

Sistematika pada penulisan laporan penelitian tugas akhir ini adalah sebagai berikut:

- BAB I : Pendahuluan
Menjelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika dan relevansi.
- BAB II : Tinjauan Pustaka
Menjelaskan tentang beberapa dasar teori yang menjadi acuan penunjang pelaksanaan penelitian tugas akhir.
- BAB III : Perancangan dan Implementasi Sistem
Menjelaskan tentang urutan perancangan sistem dan implementasi sistem yang digunakan pada penelitian tugas akhir.
- BAB IV : Pengujian dan Analisa Data
Menjelaskan mengenai hasil pengujian sistem yang dilakukan untuk mendapatkan rekomendasi sebagai tujuan akhir dari penelitian.
- BAB V : Penutup
Merupakan penutup dari laporan tugas akhir yang berisi kesimpulan dan rekomendasi yang didapatkan dari keseluruhan penelitian tugas akhir.

1.7 Relevansi

Hasil yang diperoleh dari penelitian tugas akhir ini diharapkan memberikan manfaat sebagai berikut:

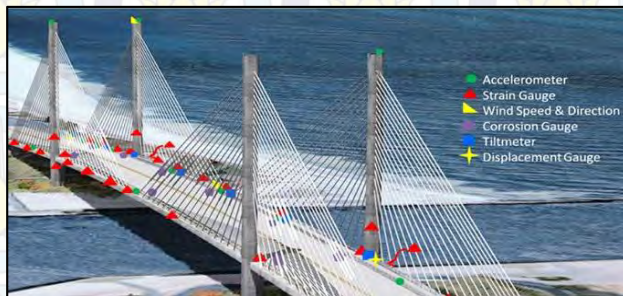
1. Sebagai sistem komunikasi data alternatif pada sistem pengawasan *Structural Health Monitoring System* yang digunakan pada Jembatan bentang panjang Surabaya-Madura.
2. Sebagai referensi untuk pengembangan *wireless sensor network* pada penelitian selanjutnya.

BAB 2

TINJAUAN PUSTAKA

2.1 Structural Health Monitoring System (SHMS)

Structural Health Monitoring System (SHMS) adalah sebuah sistem monitoring kesehatan struktur bangunan berbasis teknologi yang terdiri dari berbagai jenis sensor yang dapat memberikan data-data nyata dari elemen-elemen bangunan.



Gambar 2.1 *Structural Health Monitoring System* pada jembatan

Sistem ini dilakukan untuk mendeteksi perubahan yang mengindikasikan adanya penurunan kemampuan atau bahkan kerusakan pada suatu struktur bangunan. Seperti pada Gambar 2.1 dimana SHMS diterapkan pada jembatan dengan memasang sensor accelerometer, sensor strain, sensor suhu, sensor kecepatan angin dan beberapa jenis sensor lainnya. Hal itu bertujuan untuk melakukan proses monitoring pada jembatan tersebut.[1]

Secara tidak langsung, SHMS dapat memperpanjang umur suatu bangunan karena penurunan kemampuan atau bahkan kerusakan dini pada bangunan dapat dideteksi lebih awal sehingga tidak menyebabkan kerusakan yang parah. Secara umum tujuan dilakukannya *Structural Health Monitoring System* sendiri adalah:[7]

1. Menyediakan data respon dinamis dari struktur jembatan untuk verifikasi asumsi-asumsi disain yang digunakan untuk angin, gempa dan sebagainya.

2. Membuat sistem monitoring kesehatan jembatan yang andal sehingga memiliki fungsi pengecekan sendiri untuk memonitor adanya anomali didalam sistem.
3. Menyediakan data untuk analisa dan evaluasi kesehatan struktur jembatan.
4. Menyediakan data untuk memperkirakan kerusakan struktur dan penurunan performa jembatan untuk menentukan jadwal inspeksi dan pemeliharaan periodik.
5. Menyediakan data guna merubah tingkat keamanan lalu lintas yang disebabkan oleh gempa dan badai.
6. Menyediakan data untuk memperkirakan keandalan struktur dan arus lalu lintas paska gempa dan badai.

Secara umum sistem ini memanfaatkan sensor-sensor yang berfungsi untuk mendapatkan data dari suatu objek yang diteliti. Data-data tersebut kemudian dikirim menuju suatu pusat data dengan menggunakan sistem jaringan sensor nirkabel. Data yang telah dikirim menuju pusat data tersebut kemudian terhubung melalui suatu jaringan dengan pusat monitoring. Sehingga data-data yang diperoleh melalui sensor-sensor tersebut dapat diterima oleh pusat monitoring.

Sensor yang digunakan pada proses monitoring jembatan bergantung pada parameter apa yang dilakukan saat proses monitoring. Seperti sensor accelerometer untuk mendeteksi perubahan getaran pada jembatan, sensor wind speed untuk mendeteksi kecepatan arah angin, sensor strain untuk mendeteksi penurunan objek dan berbagai macam sensor lainnya

2.2 Jembatan Surabaya-Madura

Jembatan Nasional Suramadu adalah jembatan yang melintasi Selat Madura, menghubungkan Pulau Jawa (Surabaya) dan Pulau Madura (Bangkalan), Indonesia. Dengan panjang 5.438 m dan lebar 30m, jembatan ini merupakan jembatan terpanjang di Indonesia saat ini. Jembatan Suramadu terdiri dari tiga bagian yaitu jalan layang, jembatan penghubung dan jembatan utama.[1]

Jembatan ini diresmikan awal pembangunannya oleh Presiden Megawati Soekarnoputri pada 20 Agustus 2003 dan diresmikan pembukaannya oleh Presiden Susilo Bambang Yudhoyono pada 10 Juni 2009. Pembangunan jembatan ini ditujukan untuk mempercepat pembangunan di Pulau Madura, meliputi bidang infrastruktur dan ekonomi di Madura.

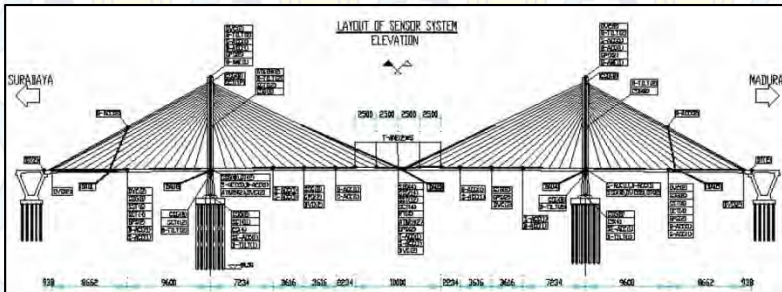
Jembatan Suramadu sudah menerapkan SHMS untuk monitoring. Beberapa macam sensor dipasang pada badan jembatan dan bagian lain jembatan untuk memberikan informasi kondisi jembatan yang terhubung dengan pusat monitoring sehingga dapat diamati secara *realtime*.



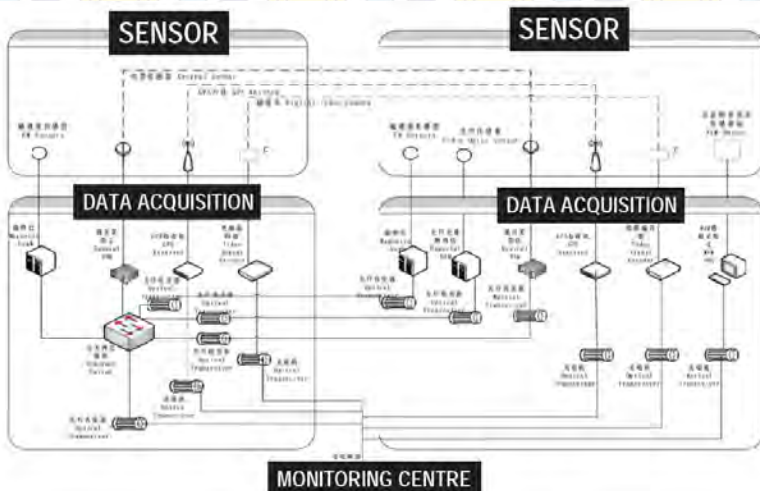
Gambar 2.2 SHMS Jembatan Suramadu

Sensor yang terpasang pada jembatan suramadu terdiri atas beberapa macam sensor diantaranya:[1]

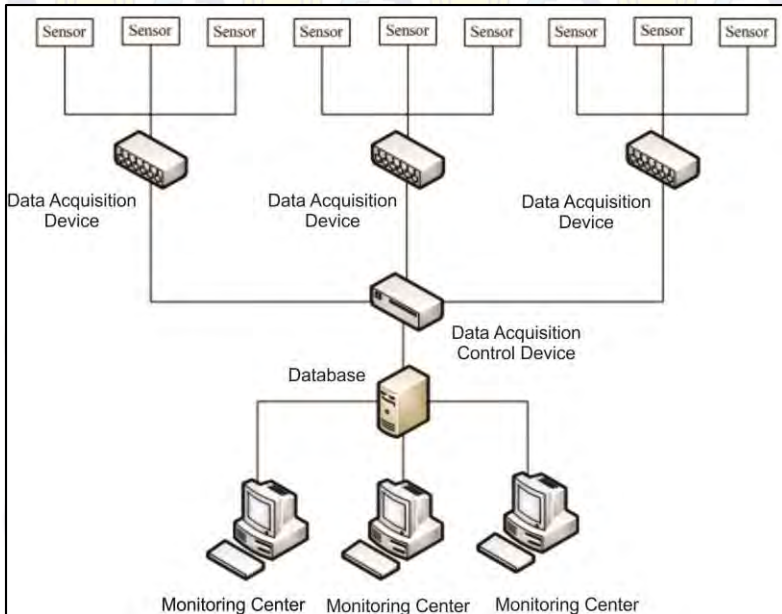
1. *Tiltmeter Sensor*
2. *Accelerometer Sensor*
3. *Strain Sensor*
4. *Weight in Motion (WIM) Sensor*
5. *Temperature Sensor*
6. *Anemometer Sensor*
7. *GPS (Global Positioning System)*
8. *Electromagnetic Sensor*



Gambar 2.3 Penempatan Sensor SHMS Jembatan Suramadu[1]



Gambar 2.4 Backbone jaringan SHMS Jembatan Suramadu[1]



Gambar 2.5 Topologi jaringan pada SHMS Jembatan Suramadu[1]

Pada SHMS Jembatan Suramadu transmisi data dikirim melalui media transmisi fiber optik. Seperti pada gambar 2.2 tentang *backbone* jaringan SHMS Suramadu, dimana akses dari sensor hingga monitoring centre berupa media fiber optik.

Topologi jaringan yang digunakan pada SHMS Suramadu menggunakan topologi jaringan tree dengan semua data terpusat pada *Data Acquisition Control Device* untuk kemudian terhubung pada *database* dan *monitoring center*.

2.3 Wireless Sensor Network (WSN)[6]

Wireless Sensor Network (WSN) merupakan suatu jaringan sensor yang terhubung secara nirkabel. Jaringan sensor sendiri merupakan suatu kumpulan sensor yang dapat merasakan (*sensing*), memproses (*processing*) serta berkomunikasi (*communicating*). dan apabila dihubungkan satu sama lain akan membentuk suatu jaringan yang dapat berfungsi untuk pengawasan terhadap suatu obyek secara kolektif.[6]

WSN dapat digunakan dalam berbagai disiplin ilmu. Pada awal penggunaannya, WSN digunakan untuk keperluan dalam dunia militer seperti pengawasan wilayah kekuasaan dan tracking musuh. Namun pada perkembangannya, WSN diaplikasikan pada banyak bidang di kehidupan sehari-hari seperti monitoring kesehatan struktur bangunan, monitoring kesehatan lingkungan dan manusia.

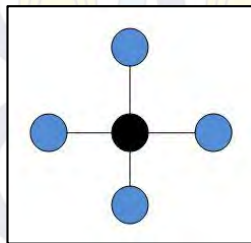
Pada perkembangannya WSN banyak diaplikasikan untuk monitoring suatu lingkungan dengan menggunakan frekuensi rendah. Salah satu contohnya adalah monitoring jembatan. Berbagai macam sensor terhubung secara nirkabel dengan jaringan dan memberikan informasi data kesehatan struktur jembatan tersebut.

WSN memiliki karakteristik khusus. WSN memerlukan konsumsi daya yang rendah dan fleksibel dalam proses monitoring dan akuisisi data. Hal itu membuat WSN lebih unggul jika dibandingkan dengan penggunaan media kabel.

Ada beberapa topologi jaringan untuk mengkoordinasikan wireless sensor network, diantaranya adalah sebagai berikut:[6]

1. Topologi Star

Topologi ini merupakan topologi paling dasar, dimana setiap node mempertahankan satu jalur komunikasi langsung dengan *gateway*. Topologi ini sederhana, namun membatasi jarak keseluruhan yang dapat dicapai.

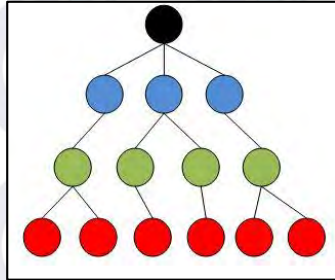


Gambar 2.6 Topologi jaringan *Star*

2. Topologi Tree

Setiap node masih mempertahankan satu jalur komunikasi untuk *gateway*, perbedaannya menggunakan node-node lain dalam mengirimkan data, namun masih dalam satu jalur tersebut. Kelemahan untuk topologi ini adalah jika node router turun, maka

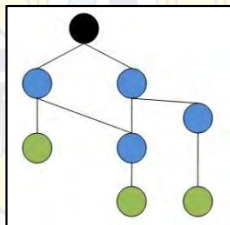
semua node yang bergantung pada node router akan kehilangan komunikasi ke gateway.



Gambar 2.7 Topologi jaringan *Tree*

3. Topologi Mesh

Topologi menggunakan jalur komunikasi yang lebih banyak untuk meningkatkan kehandalan sistem. Dalam sebuah jaringan mesh, node mempertahankan jalur komunikasi untuk kembali ke gateway, sehingga jika salah satu node router turun, secara otomatis router data akan dilewatkan melalui jalur yang berbeda. Kelemahan pada topologi ini adalah adanya delay, karena data harus melalui beberapa hop sebelum mencapai gateway.

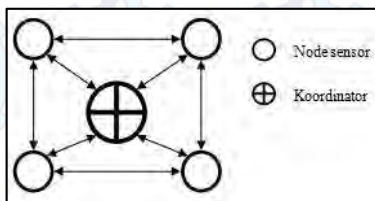


Gambar 2.8 Topologi jaringan *Mesh*

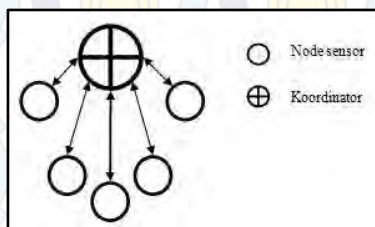
2.3.1 *Wireless Sensor Network Single-hop*

Jaringan sensor nirkabel sendiri memiliki dua jenis *routing* pengiriman data yaitu *single-hop* dan *multi-hop*. Routing *single-hop* adalah jenis routing dimana dari satu node sensor data langsung dikirimkan menuju ke router ataupun coordinator. Sedangkan untuk

routing multi-hop data dari node sensor dapat melewati beberapa node terlebih dahulu sebelum diterima oleh router atau coordinator.



Gambar 2.9 Jaringan *multi-hop*



Gambar 2.10 Jaringan *single-hop*

Pada jaringan multi-hop umumnya digunakan pada jaringan bergerak ad-hoc dan memiliki jenis *routing* pengiriman data yang bermacam-macam. Sedangkan pada tipe jaringan single-hop umumnya digunakan pada jaringan *fixed* (tetap).

2.4 Standar Komunikasi Nirkabel[6]

Standar yang menjadi pedoman pada bidang teknik elektro adalah standar IEEE. IEEE (*Institute of Electrical and Electronic Engineers*) adalah sebuah institusi internasional dimana para ahli, akademisi dan ilmuwan di bidang teknik elektro merumuskan standar-standar teknologi dalam semua aspek baik industri maupun rekayasa ilmiah yang mencakup bidang telekomunikasi, jaringan komputer, kelistrikan, astronomi serta elektronika.

Penelitian tugas akhir ini menggunakan dua standar IEEE sebagai standar protokol komunikasi data. Pertama standar IEEE 802.15.4 dan kedua standar IEEE 802.11. Standar pertama merupakan standar protokol pada WPAN (*Wireless Personal Area Network*) sedangkan standar IEEE

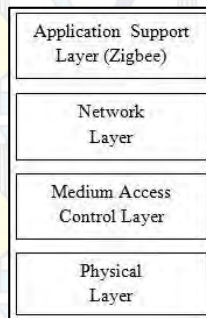
802.11 adalah standar protokol untuk WLAN (*Wireless Local Area Network*).

2.4.1 Standar IEEE 802.15.4 (ZigBee)[7]

Zigbee merupakan teknologi komunikasi nirkabel dengan data rate rendah dengan konsumsi daya yang juga rendah. zigBee menjadi sebuah nama dagang setelah alliansi zigBee dan IEEE bergabung.

Secara umum prinsip kerja dari zigbee memanfaatkan kelebihan physical radio dari standar IEEE 802.15.4. Kemudian ditambahkan dengan jaringan logika, keamanan dan aplikasi perangkat lunak.

Protokol yang digunakan pada zigbee adalah gabungan antara protokol IEEE 802.15.4 dengan protokol alliansi zigbee. Pada layer satu dan layer dua menggunakan physical layer dan media access control layer (IEEE 802.15.4) sedangkan layer tiga dan layer empat menggunakan *network layer* dan *framework application*.



Gambar 2.11 Arsitektur Protokol ZigBee

ZigBee memiliki karakteristik khusus yang membedakan dengan beberapa nama dagang yang juga menggunakan standar 802.15.xx sebagai standar protokol seperti bluetooth. ZigBee dirancang untuk komunikasi data dengan jarak dekat, konsumsi daya kecil serta kecepatan rendah (sekitar 250 kbps).

Karakteristik khusus dari ZigBee tersebut sangat cocok dengan penerapan sistem WSN, dimana sensor node yang ada pada WSN berkomunikasi dalam jarak dekat, tidak memerlukan daya tinggi serta tidak memerlukan kecepatan data yang tinggi.

a) *Physical Layer*

Physical Layer merupakan layer yang mengatur tentang bentuk interface yang berbeda-beda dari sebuah media transmisi. Spesifikasi yang berbeda misal konektor, pin, penggunaan pin, arus listrik yang lewat, encoding, sumber cahaya dan lain lain.

b) *MAC Layer*

Layer ini berfungsi untuk mengatur pengiriman data dari interface yang berbeda. Semisal pengiriman data dari ethernet 802.3 menuju ke *High-level Data Link Control (HDLC)*, pengiriman data WAN.

c) *Network Layer*

Layer ini mendefinisikan mengenai jalur (*routing*) dan pengiriman data dari ujung ke ujung. Selain itu layer ini juga melakukan pengalamatan.

d) *Aplication Layer*

Aplikasi yang saling berkomunikasi antar komputer. Aplikasi layer mengacu pada pelayanan komunikasi pada suatu aplikasi.

2.4.2 Standar IEEE 802.11

Jaringan WLAN (*Wireless Local Area Network*) merupakan pengimplementasian dari standar protokol IEEE 802.11 yang dikembangkan pada tahun 1997. Pada awalnya standar ini menggunakan transmisi data dengan kecepatan hingga 2Mbps. Dengan semakin berkembangnya teknologi, standar protokol IEEE 802.11 pun mengalami perbaikan dan pembaruan.

Pada tahun 1999 IEEE memperkenalkan dua standar protokol yang menjadi penerus dari standar IEEE 802.11 yaitu standar IEEE 802.11a dan 802.11b. Standar IEEE 802.11a menggunakan OFDM (*Orthogonal Frequency Division Multiplexing*) yang bekerja pada frekuensi 5GHz dengan kecepatan transmisi data hingga 54Mbps. Sedangkan standar IEEE 802.11b menggunakan modulasi DSSS (*Direct Sequence Spread Spectrum*) dengan frekuensi 2,4 GHz dengan kecepatan transmisi data 11Mbps.

Kemudian pada tahun 2003 diperkenalkan standar IEEE 802.11g. Standar ini menggunakan modulasi OFDM namun beroperasi pada frekuensi 2,4 GHz dengan kecepatan transmisi data hingga 5,4 Mbps.

Tabel 2.1 Jenis standar WLAN

Spesifikasi	Kecepatan	Frekuensi
802.11a	54 Mbps	5 GHz
802.11b	11 Mbps	2,4 GHz
802.11g	54 Mbps	2,4 GHz
802.11n	100 Mbps	2,4 GHz

Secara keseluruhan IEEE telah mengeluarkan lima standar protokol yang membahas tentang *wireless local area network* yaitu:

1. IEEE 802.11

IEEE 802.11 adalah sebuah standar untuk implementasi jaringan nirkabel pada frekuensi band 2,4, 3,6 dan 5 GHz. dengan adanya standar dimaksudkan agar setiap perangkat nirkabel yang berbeda tetap dapat berkomunikasi meski berbeda vendor.

2. IEEE 802.11a

IEEE 802.11a adalah sebuah teknologi jaringan nirkabel yang merupakan pengembangan lebih lanjut dari standar IEEE 802.11 namun bekerja pada bandwidth 5.8 GHz dengan kecepatan maksimum hingga 54 Mb/s. Metode transmisi yang digunakan adalah Orthogonal Frequency Division Multiplexing (OFDM), yang mengizinkan pentransmisian data secara paralel di dalam sub-frekuensi. Penggunaan OFDM memiliki keunggulan resistansi terhadap interferensi dengan gelombang lain, dan tentunya peningkatan throughput.

3. IEEE 802.11b

IEEE 802.11b merupakan pengembangan dari standar IEEE 802.11, bertujuan untuk meningkatkan kecepatan hingga 5.5 Mb/s atau 11 Mb/s tapi tetap menggunakan frekuensi 2.45 GHz. Dikenal juga dengan **IEEE 802.11 HR**. Pada prakteknya, kecepatan maksimum yang dapat diraih oleh standar IEEE 802.11b mencapai 5.9 Mb/s pada protokol TCP, dan 7.1 Mb/s pada protokol UDP. Metode transmisi yang digunakannya adalah Direct Sequence Spread Spectrum (DSSS).

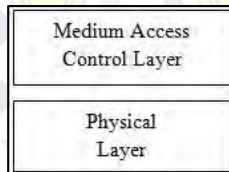
4. IEEE 802.11g

IEEE 802.11g adalah sebuah standar jaringan nirkabel yang bekerja pada frekuensi 2,45 GHz dan menggunakan metode modulasi OFDM. 802.11g yang dipublikasikan pada bulan Juni

2003 mampu mencapai kecepatan hingga 54 Mb/s pada pita frekuensi 2,45 GHz, sama seperti halnya IEEE 802.11 biasa dan IEEE 802.11b. Standar ini menggunakan modulasi sinyal OFDM, sehingga lebih resistan terhadap interferensi dari gelombang lainnya.

5. IEEE 802.11n

IEEE 802.11n adalah standar jaringan yang bekerja pada frekuensi 2,4 GHz dan kecepatan transfer datanya mencapai 100-200 Mbps. Standar ini dirancang untuk memperbaiki fitur 802.11g dalam jumlah bandwidth yang didukung dengan memanfaatkan beberapa sinyal nirkabel dan antenna (MIMO).



Gambar 2.12 Arsitektur Protokol WLAN

Arsitektur protokol yang digunakan pada keluarga standar IEEE 802.11 adalah dua layer yaitu physical layer dan medium access control layer.

2.5 Accelerometer

Sensor accelerometer adalah sebuah sensor yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran. Accelerometer juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan instalasi pengamanan.[1]

Accelerometer yang diletakan di permukaan bumi dapat mendeteksi percepatan persatuan g pada titik vertikalnya, untuk percepatan yang dikarenakan oleh pergerakan horizontal maka accelerometer akan mengukur percepatannya langsung ketika bergerak secara horizontal.

Parameter penting pada accelerometer adalah sensitivitas, ukuran seberapa banyak perubahan yang terjadi pada hasil output sensor berdasarkan perubahan percepatan. Satuan dari sensitivitas adalah volts/g. Menurut tipenya, sensor accelerometer terbagi dalam beberapa tipe yaitu:

1. *Capacitive*

Lempengan metal pada sensor memproduksi sejumlah

kapasitansi, perubahan kapasitansi akan mempengaruhi percepatan

2. *Piezoelectric*

Kristal piezoelectric yang terdapat pada accelerometer jenis ini mengeluarkan tegangan yang selanjutnya dikonversi menjadi percepatan

3. *Piezoresistive*

Lempengan yang secara resistan akan berubah sesuai dengan perubahan percepatan

4. *Hall effect*

Percepatan yang dirubah menjadi sinyal elektrik dengan cara mengukur setiap perubahan pergerakan yang terjadi pada daerah yang terinduksi magnet.

5. *Magnetoresistive*

Perubahan percepatan diketahui berdasarkan resistivitas material karena adanya daerah yang terinduksi magnet

6. *Heat Transfer*

Percepatan dapat diketahui dari lokasi sebuah benda yang dipanaskan dan diukur ketika terjadi percepatan dengan sensor temperature.



Gambar 2.13 Accelerometer

MMA7361 adalah salah satu jenis sensor accelerometer yang mempunyai 3-axis arah sensor dan menangkap data berupa getaran pada suatu objek. Penggunaan sensor accelerometer pada tugas akhir ini karena data yang diambil pada jembatan berupa data getaran jembatan dan hal tersebut sesuai dengan kegunaan dari sensor accelerometer.

2.6 Arduino[9]

Arduino merupakan suatu perangkat elektronik yang bersifat *open-source*. Mikrokontroler yang ada pada board arduino diprogram menggunakan bahasa pemrograman arduino.

Platform arduino terdiri dari arduino board, shield, bahasa pemrograman arduino dan arduino development environment. Shield pada arduino berfungsi untuk menambah kemampuan dari arduino.

Shield arduino memiliki berbagai macam jenis dan kemampuan yang berbeda. Bahasa pemrograman pada arduino hampir sama dengan pemrograman C. Arduino development environment adalah perangkat lunak yang digunakan untuk menulis dan meng-compile program pada arduino. Arduino development environment juga digunakan untuk meng-*upload* program yang telah di-*compile* ke memori arduino board.

Penggunaan arduino pada komunikasi data wireless sensor network karena kelebihan dari mikrokontroler arduino seperti kemudahan untuk *coding* dan *compile*. Selain itu penggunaan bahasa arduino lebih mudah dipahami karena memiliki interface yang memudahkan pengguna.

2.6.1 Arduino Uno

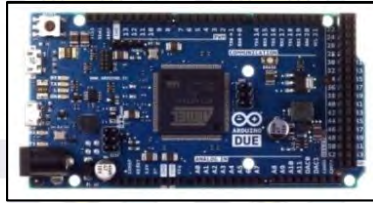
Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. IC (integrated circuit) ini memiliki 14 input/output digital (6 output untuk PWM), 6 analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset.



Gambar 2.14 Arduino Uno

2.6.2 Arduino Due

Arduino Due adalah sebuah papan mikrokontroler berbasis Atmel SAM3X8E ARM Cortex-M3 CPU. Arduino Due merupakan papan mikrokontroler arduino pertama yang menggunakan 32-bit ARM. Arduino Due dapat digunakan pada seluruh perangkat shield yang bekerja pada 3.3 Volt.



Gambar 2.15 Arduino Due

2.6.3 Arduino Ethernet Shield

Arduino Ethernet Shield adalah salah satu perangkat yang menghubungkan papan mikrokontroler dengan internet. Arduino Ethernet Shield berbasis Wiznet W5100 ethernet chip.



Gambar 2.16 Arduino Ethernet Shield

Pemrograman dilakukan pada library ethernet arduino. Pada Ethernet Shield tersedia konektor RJ-45 yang langsung terhubung dengan daya dari ethernet shield. Selain itu juga tersedia slot micro-SD sebagai media penyimpan data eksternal. Pemrograman pada pola penyimpanan eksternal dapat dilakukan pada SD-library.

2.6.4 Arduino Xbee Shield[8]

Arduino Xbee Shield menyederhanakan tugas interfacing sebuah Xbee dengan Arduino. Shield dapat dipasang dengan Arduino Pro ataupun USB, dan dilengkapi dengan kemampuan komunikasi nirkabel menggunakan modul Xbee.



Gambar 2.17 Arduino Xbee Shield

2.7 *Xbee*[8]

Xbee adalah perangkat *wireless transmitter-receiver* yang mengimplementasikan standar protokol IEEE 802.15.4. Pada sistem *wireless sensor network*. *Xbee* sering digunakan karena memiliki konsumsi daya rendah serta kecepatan transmisi rendah.

Pada penggunaan *Xbee*, terdapat beberapa hal yang harus diperhatikan, yaitu:

1. *Clear Channel Assesment (CCA)*
Berfungsi untuk mengidentifikasi frekuensi yang akan digunakan pada saat komunikasi data. Frekuensi sibuk atau tidak.
2. *Addressing*
Berfungsi untuk pengalamatan pada 16bit yang akan digunakan pada jaringan komunikasi data.
3. *Error Checking and Acknowledge*
Berfungsi sebagai kontrol dan pengecekan pada proses komunikasi data.

Networking pada *Xbee* memiliki kemampuan untuk mengirim data pada alamat-alamat yang telah ditentukan. Karena pada *Xbee* dapat diatur dengan alamat alamat yang berbeda pada tiap-tiap modul *Xbee*. Dari pengalamatan tersebut memungkinkan data dikirim hanya ke sebuah modul, atau ke kelompok modul dan dapat menghindari kesalahan dan tumpukan pengalamatan.



Gambar 2.18 *Xbee*

Pada penelitian tugas akhir ini menggunakan Xbee pro series1. *Xbee* tipe ini mampu berkomunikasi memenuhi untuk komunikasi *point-to-point* dan *point-to-multipoint*.

Beberapa Keuntungan yang diperoleh dari penggunaan zigbee adalah:

1. Mendukung untuk topologi multiple jaringan: static, dynamic, star dan mesh.
2. Dapat menangani jaringan hingga 64.000 node
3. Mempunyai umur baterai yang cukup panjang

2.8 NanoStation M2

Jalur akses nirkabel untuk membuat daya tinggi koneksi Wi-Fi di kejauhan hingga 5 km atau lebih. Dalam modul radio desain kapasitas kompak terpadu dari 400 MW dan bipanelnya antena dengan sudut 60 radiasi dipita 2,4 GHz. Fungsi sebagai jalur akses, jembatan, dan repeater.



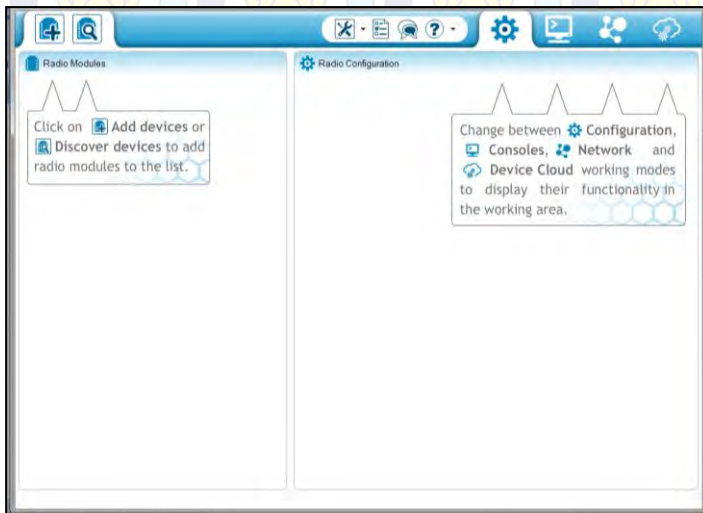
Gambar 2.19 NanoStation M2

2.9 XCTU

XCTU adalah sebuah aplikasi yang disediakan oleh Digi. Program ini didesain untuk produk RF Digi dan juga mempermudah penggunaan dengan *GUI*.

XCTU didesain dapat dioperasikan dengan segala jenis komputer yang berbasis *windows* maupun *macOS*. XCTU dapat diunduh secara bebas dari *website* resmi Digi.

Fasilitas yang terdapat pada XCTU selain memudahkan untuk *setting* modul *Xbee* juga dapat menampilkan data data yang dikirim dan diterima dari komunikasi menggunakan *Xbee*.



Gambar 2.20 Tampilan *XCTU*

Pada XCTU untuk pengaturan modul *Xbee* dapat dilakukan dengan mudah. Karena kita dapat memasukkan secara langsung parameter-parameter yang akan kita gunakan. Ada beberapa parameter yang perlu kita atur untuk *Xbee* yang akan kita gunakan agar dapat berkomunikasi sesuai dengan yang kita inginkan. Parameter tersebut adalah :

- ID (PAN ID), merupakan alamat dari PAN (*Personal Area Network*) ID dari modul RF.
- DL (*destination address low*)

- DH (*destination address high*)
- SH (*serial number high*)
- SL (*serial number low*)
- CH (*chanel*), merupakan pemilihan chanel dimana komunikasi dilakukan oleh modul RF.
- MY, merupakan parameter yang menunjukkan alamat dari modul RF.

2.10 Webservice

WebServer adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

HTTP sendiri adalah protokol komunikasi antara web server dengan *client*. Dengan adanya protokol ini, komunikasi antar web server dengan client dapat saling dimengerti dan lebih sederhana.

Proses yang dimulai dari permintaan web client (browser), diterima web server, kemudian diproses dan dikembalikan hasil prosesnya oleh web server ke web client. Secara umum, web server hanya memproses tiap masukan yang berasal dari web client.

2.10.1 Webservice Apache

Apache merupakan salah satu jenis webserver yang banyak digunakan. Program ini pertama kali didesain untuk sistem operasi UNIX. Apache webserver merupakan program aplikasi yang berjalan di server, yang berfungsi untuk menjalankan aplikasi web sehingga bisa diakses oleh client baik melalui jaringan intranet maupun jaringan internet.

Nama Apache diambil dari kata “A Patchy Server” yang memiliki arti server yang penuh perbaikan. Saat ini apache dipergunakan secara luas. Hal itu disebabkan karena bersifat *opensource* dengan kinerja relatif stabil.

Webserver apache memiliki kelebihan dari beberapa pertimbangan penjelasan sebelumnya, diantaranya:

1. Apache termasuk kategori freeware
2. Proses instalasi yang mudah
3. Dapat dioperasikan pada berbagai jenis sistem operasi
4. Konfigurasi yang mudah dan sederhana

2.10.2 Database SQL

SQL yang merupakan singkatan dari Structured Query Language *adalah bahasa pemrograman khusus yang digunakan untuk manajemen data dalam*. SQL biasanya berupa perintah sederhana yang berisi instruksi-instruksi untuk manipulasi data. Perintah SQL ini sering juga disingkat dengan sebutan '**query**'.

Perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat 3 jenis perintah dasar SQL :

- Data Definition Language (DDL) adalah jenis instruksi SQL yang berkaitan dengan pembuatan struktur tabel maupun database. Termasuk diantaranya : CREATE, DROP, ALTER, dan RENAME.
- Data Manipulation Language (DML) adalah jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaimana menginput, menghapus, memperbaharui serta membaca data yang tersimpan di dalam database. Contoh perintah SQL untuk DML : SELECT, INSERT, DELETE, dan UPDATE.
- Data Control Language (DCL) adalah jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (user) yang dapat mengakses database maupun tabel. Termasuk diantaranya : GRANT dan REVOKE.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL yang multithread dan multiuser. Tidak seperti perangkat lunak apache yang dikembangkan oleh komunitas umum, hak cipta untuk koding dimiliki oleh penulisnya masing-masing.

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu command line. Selain itu juga terdapat perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yaitu phpMyAdmin.

2.10.3 Pemrograman PHP

PHP yang merupakan singkatan dari Personal Home Page Hypertext Preprocessor adalah bahasa pemrograman script yang banyak digunakan. PHP sering digunakan untuk memprogram situs web. PHP dapat dilihat sebagai pilihan lain dari berbagai jenis pemrograman web lainnya seperti C, JSP, CGI, dan lain lain.

PHP disebut sebagai sebuah server side embedded script language artinya sintak-sintak dan perintah yang diberikan akan dijalankan oleh server. Sehingga bahasa pemrograman PHP akan dieksekusi oleh server dan yang dikirimkan ke server adalah hasil jadi dalam bentuk HTML. Aplikasi pada PHP pada umumnya akan memberikan hasil pada webbrowser, namun prosesnya secara keseluruhan tetap dijalankan di server.

Ketika menggunakan PHP sebagai server side embedded script language maka server akan melakukan hal-hal sebagai berikut:

1. Membaca permintaan dari client
2. Mencari halaman di server
3. Melakukan instruksi yang diberikan PHP untuk melakukan perubahan pada halaman
4. Mengirim kembali halaman tersebut kepada client baik melalui intranet maupun internet

2.11 Parameter Untuk Kerja Jaringan

Pengukuran unjuk kerja suatu jaringan dapat dilakukan dengan pengamatan terhadap beberapa parameter-parameter penting yang menunjukkan kinerja suatu jaringan. Parameter tersebut diantaranya *packet loss*, *throughput* dan *delay end-to-end*.

2.11.1 Packet Loss

Packet loss merupakan kegagalan transmisi pada suatu paket data untuk mencapai tujuannya. Kegagalan tersebut dapat disebabkan oleh beberapa kemungkinan seperti terjadinya *overload* trafik dalam jaringan, kerusakan pada media fisik, dan lain lain. Secara matematis *packet loss* dapat dituliskan dengan persamaan:

$$\text{Packet Loss} = \frac{\text{Paket dikirim} - \text{Paket diterima}}{\text{Paket dikirim}} \times 100\% \quad (1.1)$$

2.11.2 Throughput

Throughput adalah jumlah data per unit waktu yang dikirimkan ke penerima tertentu dalam suatu jaringan, dari *node* jaringan, atau dari satu *node* ke *node* lain melalui *node* komunikasi. *Throughput* biasanya diukur dalam satuan bit per detik atau bps. Sistem *throughput* atau *aggregate throughput* adalah jumlah dari kecepatan data yang dikirim kesemua terminal dalam sebuah jaringan. *Throughput* maksimum sebuah

node atau *link* komunikasi didefinisikan sebagai *throughput* asimtotik ketika beban sangat besar. Beberapa faktor yang dapat mempengaruhi *throughput* adalah :

1. Perangkat pada jaringan
2. Topologi jaringan
3. Jumlah pengguna pada jaringan
4. Tipe data yang dikirim

Secara matematis *throughput* dituliskan sebagai berikut.

$$\eta = \frac{N}{\tau} \times 8(bps) \quad (1.2)$$

Dengan :

- η = *Throughput*
 N = Paket data
 τ = Total waktu untuk transmisi

2.11.3 Delay

Delay adalah selang waktu yang diperlukan oleh data yang dikirim dari *end device* hingga mencapai *coordinator*. Faktor-faktor yang mempengaruhi *end-to-end delay* adalah waktu untuk menemukan *route* pada jaringan, adanya *delay* pada eksekusi dari data yang masuk ke *router*, dan kondisi dari media tranmisi data. *End-to-end delay* di dapat nilainya dengan cara menghitung selisih waktu pengiriman data dan waktu saat data diterima di tujuan. Secara matematis dapat dituliskan dengan persamaan berikut.

$$\Delta t = t_1 - t_0 \quad (1.3)$$

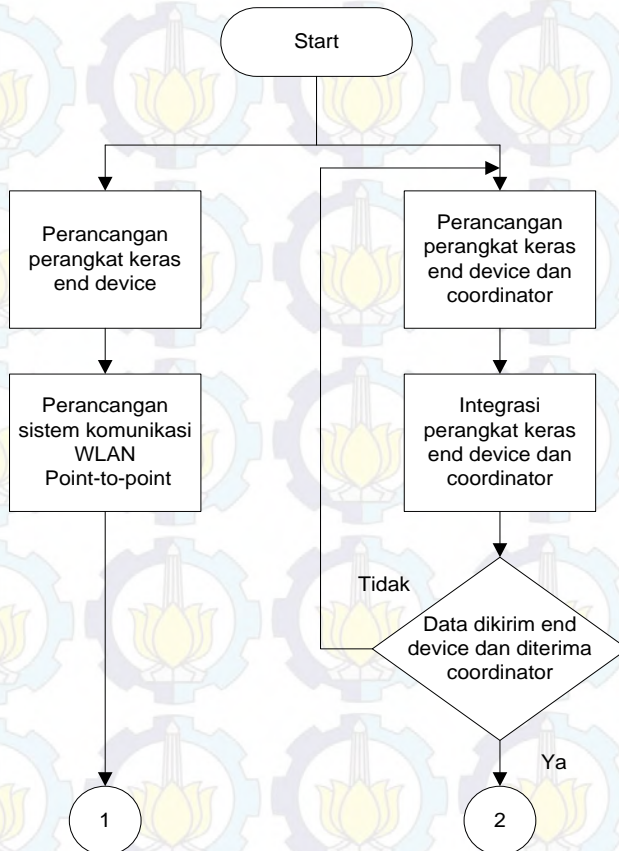
Dengan:

- Δt = *end-to-end delay*
 t_t = waktu terima
 t_o = waktu kirim

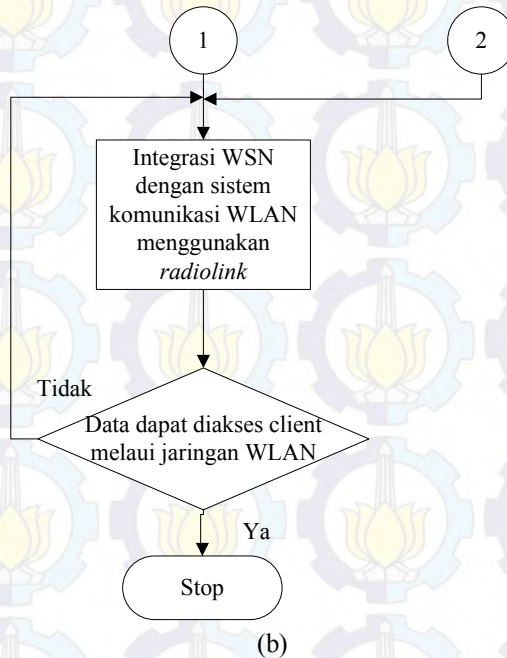
BAB 3

METODE PENELITIAN

Pada bab ini menjelaskan tentang perancangan dan pengimplementasian *wireless sensor network* serta bentuk komunikasi data yang digunakan pada *structural health monitoring system* jembatan bentang panjang. Berikut adalah alur pengerjaan pada penelitian tugas akhir ini:



(a)



Gambar 3.1 Diagram alir pengerjaan tugas akhir (a) start hingga integrasi menyusun *node* (b) integrasi antar *node*

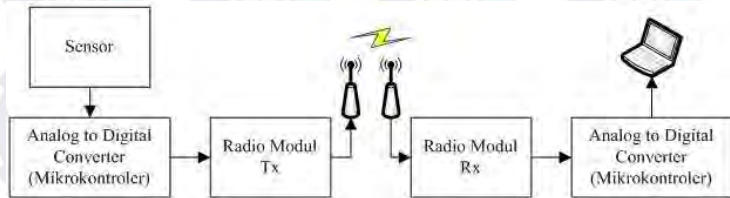
Sistem ini dirancang dengan menggunakan dua skenario. Pada skenario pertama terdapat satu node yang berfungsi sebagai pengambil data (*end device*), dan pada skenario dua terdapat dua node yang berfungsi sebagai pengambil data (*end device*) dan satu kordinator. *End device* akan melakukan pengiriman data, sedangkan kordinator akan melakukan penerimaan data.

3.1 Aplikasi *Wireless Sensor Network* pada jembatan

Sistem monitoring kesehatan jembatan bentang panjang, terdapat dua bagian penting dalam melakukan monitoring. Bagian pertama adalah bagian *sensing* dan bagian yang kedua adalah bagian monitoring data.

Pada bagian *sensing* terdiri dari beberapa node sensor yang berfungsi untuk mengambil data dari objek yang diamati. Sensor-sensor

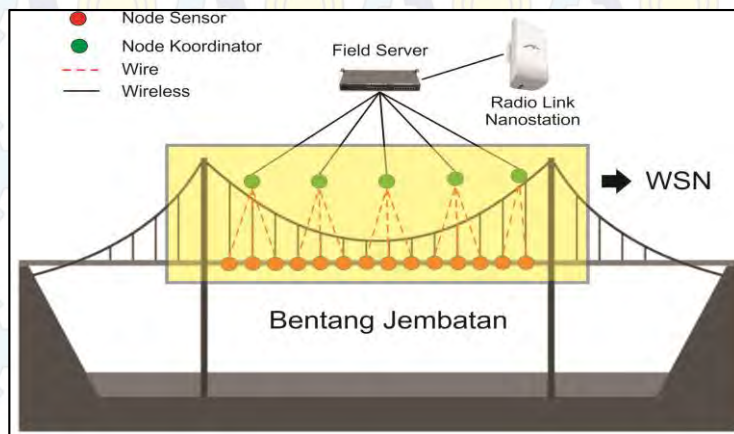
tersebut terhubung dengan mikrokontroler yang berfungsi sebagai pengolah data dan modul komunikasi. Integrasi antara sensor dan mikrokontroler disebut dengan node sensor (*end device*). Node sensor yang berfungsi sebagai *end device* terhubung dengan node koordinator. Komunikasi yang berlangsung antara sisi node sensor dengan node koordinator terjadi dengan menggunakan *wireless sensor network*.



Gambar 3.2 Blok diagram komunikasi data WSN

Node koordinator yang terdiri dari mikrokontroler dan modul komunikasi menjadi pusat dari lalu lintas komunikasi data pada sistem ini. Selain berfungsi sebagai penerima data dari node sensor, koordinator juga menjadi penghubung dengan basis data.

Selanjutnya pada bagian monitoring adalah kelanjutan dari bagian sensing, dimana data dari sensor yang terdapat pada datalogger akan diolah serta ditampilkan dalam bentuk angka, grafik dan juga gambar. Data tersebut kemudian dikirimkan menuju pusat monitoring menggunakan media kabel ataupun nirkabel.



Gambar 3.3 Arsitektur sistem komunikasi pada SHMS jembatan

Pada penelitian tugas akhir ini yang menjadi fokus penelitian adalah komunikasi data pada bagian sensing hingga gateway. Bagaimana menghubungkan node sensor dengan node kordinator secara nirkabel dan bagaimana menghubungkan kordinator dengan jaringan lokal.

Wireless sensor network diaplikasikan pada bagian sensing. Protokol yang digunakan adalah *ZigBee*. Dimana protokol *ZigBee* merupakan penggabungan dari protokol aliansi Zigbee dan IEEE 802.15.4. Tiga sensor node mengambil data getaran jembatan bentang panjang menggunakan sensor accelerometer. Data dari tiga sensor tersebut selanjutnya dikirimkan menuju satu kordinator. Jalur pengiriman data dari tiga sensor adalah single-hop yang bersifat tetap(*fixed*) tanpa mencari jalur pengiriman baru.

Kordinator tidak hanya berfungsi sebagai penerima data yang telah dikirimkan oleh sensor node. Selanjutnya kordinator terhubung dengan jaringan lokal melalui *WLAN* sehingga data yang telah diterima kordinator dapat diakses dan diolah lebih lanjut.

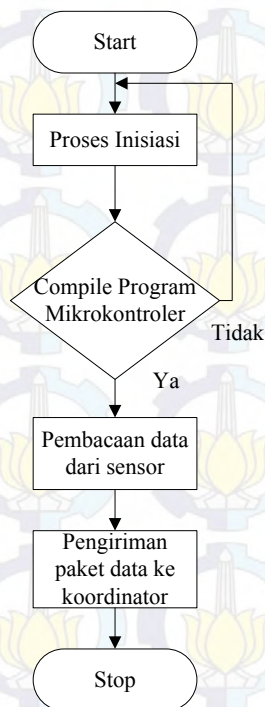
Perancangan pada bagian sensing ini meliputi perancangan perangkat keras sensor node dan kordinator serta perancangan sistem komunikasi nirkabel pada kordinator menggunakan *radio-link*. Selanjutnya seluruh perangkat keras yang telah dirancang diintegrasikan sehingga seluruh sistem komunikasi data pada bagian sensing dapat dilakukan.

3.2 Perancangan Perangkat Keras

Dalam penelitian tugas akhir ini perangkat keras terdiri dari tiga bagian, yaitu bagian sensing, bagian pengolahan data serta bagian komunikasi. Tiap bagian tersebut menggunakan perangkat keras yang berbeda sesuai dengan fungsinya masing-masing. Pada sisi node sensor penggunaan perangkat keras berbeda dengan perangkat keras pada sisi node koordinator.

3.2.1 Perancangan Node Sensor

Node sensor (*end device*) terdiri dari sensor accelerometer sebagai alat sensing, mikrokontroler Arduino uno dan due sebagai pengolah data serta *xbee* sebagai modul komunikasi untuk dapat terhubung dengan perangkat pada sisi node koordinator.



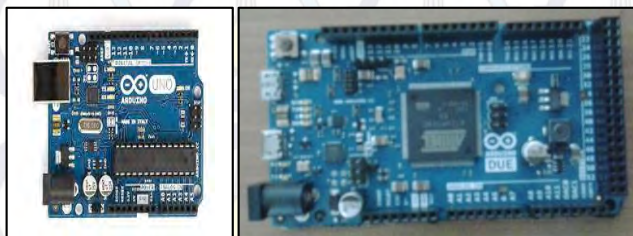
Gambar 3.4 Diagram alir node sensor (*end device*)

Pada penelitian tugas akhir ini sensor yang digunakan adalah sensor accelerometer tipe MMA 7361. Sensor ini mampu mempresentasikan tiga sumbu gerakan pada sumbu X, Y, dan Z yang telah disesuaikan dengan titik gravitasi bumi. Sensor ini memiliki *switch range* getaran yaitu ± 2 g dan ± 6 g, dimana g adalah satuan untuk percepatan gravitasi m/s^2 . Untuk penelitian ini *range* yang digunakan *range* terbesar yaitu ± 6 g.



Gambar 3.5 Sensor accelerometer MMA 7361

Mikrokontroler yang digunakan adalah mikrokontroler *Arduino uno* dan *Arduino due*. Untuk listing program dari arduino uno dan due sebagai node sensor dapat dilihat pada lampiran. Mikrokontroler disini berperan sebagai bagian pengolah data dari sensor sebelum ditransmisikan. Mikrokontroler berperan juga sebagai ADC (*analog digital converter*) dari data sensor. *Arduino* memiliki dua resolusi ADC yaitu 8 bit dengan *range* nilai digital 0 – 255, dan 16 bit dengan *range* nilai 0 – 65535.



Gambar 3.6 Arduino Uno dan Due

Mikrokontroler juga berfungsi untuk memberikan catatan waktu pengambilan data dari hasil pembacaan sensor accelerometer.

Mikrokontroler juga berfungsi untuk pengalamanan dengan memberi informasi dari node sensor (*end device*) mana data berasal. Agar dapat dikirim dengan format yang mudah mikrokontroler juga memberi *start* dan *stop* bit sebagai tanda awal dan akhir dari paket yang dikirim.



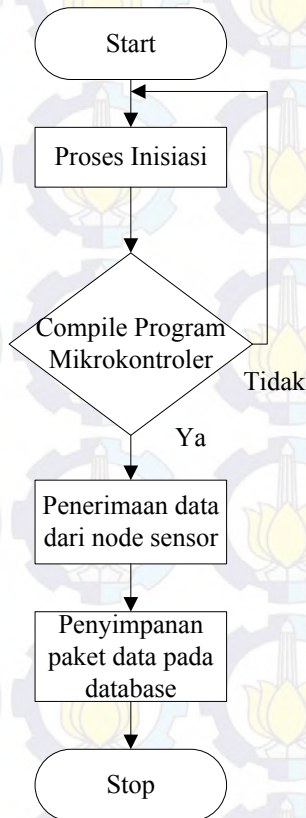
Gambar 3.7 Xbee

Pada sisi komunikasi data, menggunakan modul *xbee* sebagai modul komunikasi data. *Xbee* yang digunakan adalah *Xbee Pro Series 1*. Modul ini adalah perangkat dengan protokol standart IEEE 802.15.4. *Xbee Pro Series 1* memiliki kemampuan berkomunikasi secara *point-to-point*, dan *point-to-multipoint*. Pada bagian node sensor (*end device*) modul *xbee* diatur agar dapat melakukan komunikasi data secara *multipoint-to-point*.

3.2.2 Perancangan Node Koordinator

Node koordinator terdiri dari mikrokontroler arduino uno sebagai pengolah data serta *xbee* sebagai modul komunikasi untuk dapat berkomunikasi dengan perangkat pada sisi node sensor (*end device*).

Pada sisi koordinator tidak menggunakan ataupun terhubung dengan sensor karena pada sisi koordinator ini hanya berfungsi sebagai penerima data dari node sensor untuk kemudian diolah dan disimpan pada basis data.



Gambar 3.8 Diagram alir node koordinator

Pada node koordinator menggunakan mikrokontroler arduino uno sebagai pengolah data dan modul komunikasi xbee untuk dapat terhubung secara nirkabel dengan node sensor. Untuk listing program arduino uno sebagai node coordinator ada pada lampiran. Dan spesifikasi perangkat keras yang digunakan pada node sensor (*end device*) menggunakan arduino due.

3.3 Integrasi Perangkat Keras

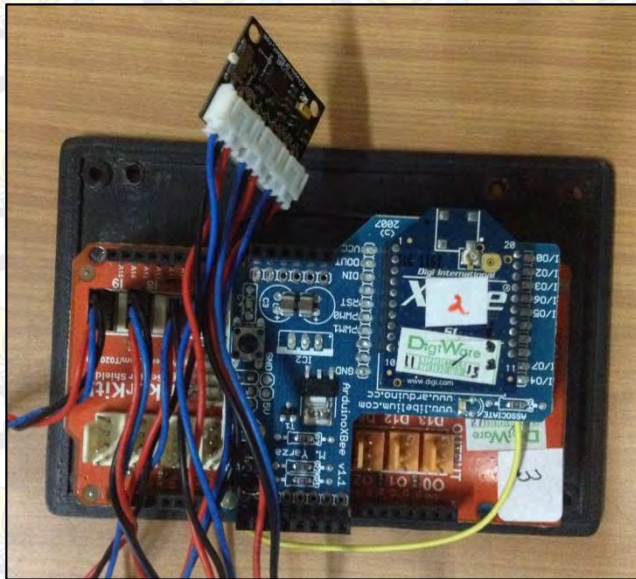
Setelah proses perancangan dan pengaturan perangkat keras baik pada node sensor maupun node koordinator, selanjutnya adalah integrasi

perangkat keras agar dapat mengimplementasikan sistem *wireless sensor network* pada jembatan bentang panjang.

3.3.1 Integrasi Perangkat Keras Node Sensor

Perangkat-perangkat keras penyusun yang dirancang untuk diimplementasikan pada node sensor terdiri dari tiga perangkat yang memiliki fungsi berbeda-beda. Ketiga perangkat tersebut digabung dalam satu sistem kerja.

Sensor accelerometer sebagai perangkat yang melakukan penginderaan terhadap objek mengalirkan data menuju arduino due yang berperan sebagai mikrokontroler.



Gambar 3.9 Perangkat node sensor

Data yang masuk kemudian diolah oleh arduino due. Data tersebut diolah hingga menjadi informasi digital yang menunjukkan hasil penginderaan dari sensor accelerometer.

Perangkat Xbee menjalankan fungsi komunikasi dimana data hasil penginderaan oleh sensor yang telah diolah menjadi informasi

digital pada arduino dikirimkan menuju node koordinator melalui frekuensi radio.

3.3.2 Integrasi Perangkat Keras Node Koordinator

Proses kerja dari perangkat keras node koordinator hampir sama dengan proses kerja pada perangkat node sensor. Namun pada koordinator perangkat keras yang digunakan hanya mikrokontroler arduino uno dan modul xbee. Sensor tidak digunakan karena pada node koordinator hanya melakukan eksekusi penerimaan data dari tiga node sensor tanpa melakukan penginderaan langsung.



Gambar 3.10 Perangkat node koordinator

Data yang dikirimkan oleh node sensor masuk dan diterima oleh modul xbee pada node koordinator. Data tersebut kemudian dialirkan menuju mikrokontroler arduino uno yang melakukan fungsi pembacaan dan pengolahan data.

Mikrokontroler arduino terhubung dengan PC melalui kabel PORT. Sehingga data yang telah diolah arduino uno langsung dikirim

menuju PC secara serial untuk penyimpanan data secara langsung pada *database*.

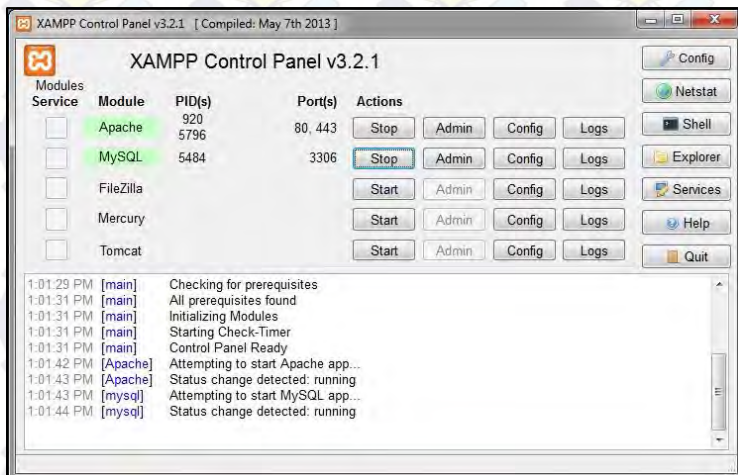
3.4 Perancangan Database dan Web

Perangkat lunak yang digunakan pada perancangan database dan web ini adalah perangkat lunak yang bersifat *opensource* antara lain XAMPP, Python dan Notepad ++.

3.4.1 Webserver

Dalam penelitian tugas akhir ini aplikasi perangkat lunak webserver yang digunakan adalah Xampp webserver. Ada beberapa keunggulan dari Xampp webserver yaitu bersifat *opensource* dan sederhana. Proses instalasi Xampp webserver cukup mudah tanpa melakukan perubahan konfigurasi. Kelebihan Xampp webserver yang lain adalah dapat digunakan pada berbagai jenis sistem operasi seperti Windows, Unix dan lainlain.

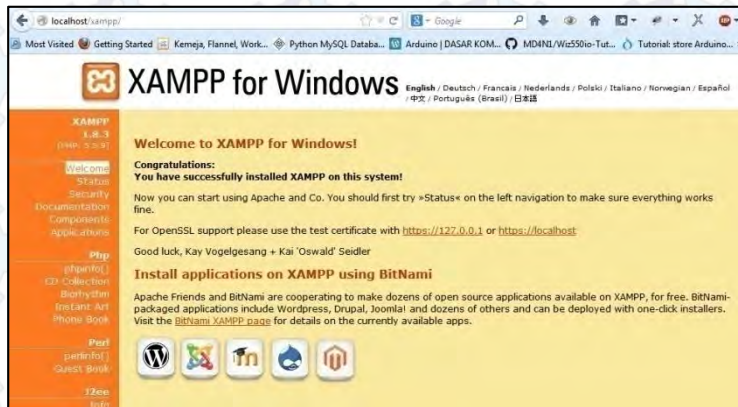
Langkah penggunaan Xampp webserver:



Gambar 3.11 Xampp control panel

Setelah melakukan instalasi, maka akan keluar tampilan halaman Xampp control panel seperti pada Gambar 3.11. Untuk menjalankan program Apache dan MySQL jalankan program dengan

melakukan start. Kemudian untuk masuk pada halaman utama Xampp melalui menu admin pada jendela Xampp control panel.



Gambar 3.12 Halaman utama Xampp

Setelah masuk pada halaman utama pada Xampp, pilih menu phpMyAdmin untuk masuk pada MySQL dan mulai membuat program database.

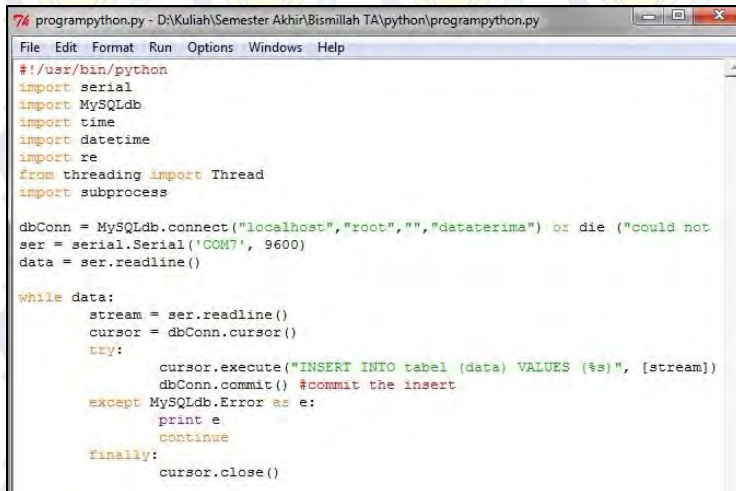


Gambar 3.13 Halaman utama phpmyadmin

Pada penelitian tugas akhir ini data yang masuk pada node koordinator adalah data serial dan berulang. Diperlukan suatu cara untuk mengirimkan data serial dari node koordinator agar dapat masuk dan disimpan dalam database. Karena data serial tidak dapat secara langsung dikirim dan disimpan.

Pemrograman menggunakan perangkat lunak python digunakan untuk menghubungkan data serial yang dikirim dari node koordinator menuju database.

Gambar adalah gambar tampilan paada pemrograman python(GUI) yang berfungsi sebagai penghubung antara node koordinator dan database.



```
7% programpython.py - D:\Kuliah\Semester Akhir\Bismillah TA\python\programpython.py
File Edit Format Run Options Windows Help

#!/usr/bin/python
import serial
import MySQLdb
import time
import datetime
import re
from threading import Thread
import subprocess

dbConn = MySQLdb.connect("localhost","root","","dataterima") or die ("could not
ser = serial.Serial('COM7', 9600)
data = ser.readline()

while data:
    stream = ser.readline()
    cursor = dbConn.cursor()
    try:
        cursor.execute("INSERT INTO tabel (data) VALUES (%s)", [stream])
        dbConn.commit() #commit the insert
    except MySQLdb.Error as e:
        print e
        continue
    finally:
        cursor.close()
```

Gambar 3.14 Tampilan perangkat lunak python (GUI)

Setelah melakukan pemrograman melalui perangkat lunak python, data serial dari node koordinator dikirim pada database. Data yang masuk pada database akan tersimpan dan terupdate secara berkala dan berulang.

Data serial yang masuk berupa alamat sensor, data getaran sensor serta *time stamp* sensor. Pada database kemudian dibuat *time stamp* agar dapat terlihat delay pengiriman dari node sensor dan node koordinator sampai data diterima oleh database.

Gambar dibawah ini adalah gambar tampilan pada database MySQL yang terdiri dari data sensor serta time stamp data yang masuk pada database.

Server: 127.0.0.1 » Basis data: test » Tabel: sensor

				id	time	waktu	X	Y	Z
Ubah	Salin	Hapus		1	2015-12-14 00:22:01	0:01:06	-12.1359	-13.4518	-15.8716
Ubah	Salin	Hapus		2	2015-12-14 00:22:01	0:01:07	-12.1344	-13.4533	-15.8727
Ubah	Salin	Hapus		3	2015-12-14 00:22:02	0:01:08	-12.1356	-13.4533	-15.8738
Ubah	Salin	Hapus		4	2015-12-14 00:22:02	0:01:08	-12.1359	-13.4533	-15.8723
Ubah	Salin	Hapus		5	2015-12-14 00:22:03	0:01:09	-12.1359	-13.4533	-15.8727
Ubah	Salin	Hapus		6	2015-12-14 00:22:04	0:01:09	-12.1359	-13.4526	-15.8720
Ubah	Salin	Hapus		7	2015-12-14 00:22:05	0:01:10	-12.1359	-13.4533	-15.8723
Ubah	Salin	Hapus		8	2015-12-14 00:22:05	0:01:11	-12.1352	-13.4537	-15.8735
Ubah	Salin	Hapus		9	2015-12-14 00:22:06	0:01:12	-12.1359	-13.4522	-15.8720
Ubah	Salin	Hapus		10	2015-12-14 00:22:07	0:01:13	-12.1359	-13.4529	-15.8716
Ubah	Salin	Hapus		11	2015-12-14 00:22:08	0:01:13	-12.1344	-13.4548	-15.8727
Ubah	Salin	Hapus		12	2015-12-14 00:22:08	0:01:14	-12.1359	-13.4552	-15.8723
Ubah	Salin	Hapus		13	2015-12-14 00:22:09	0:01:15	-12.1344	-13.4570	-15.8731
Ubah	Salin	Hapus		14	2015-12-14 00:22:10	0:01:15	-12.1352	-13.4559	-15.8712
Ubah	Salin	Hapus		15	2015-12-14 00:22:10	0:01:16	-12.1359	-13.4574	-15.8735
Ubah	Salin	Hapus		16	2015-12-14 00:22:11	0:01:17	-12.1359	-13.4578	-15.8720
Ubah	Salin	Hapus		17	2015-12-14 00:22:12	0:01:17	-12.1359	-13.4578	-15.8742
Ubah	Salin	Hapus		18	2015-12-14 00:22:12	0:01:18	-12.1359	-13.4574	-15.8731

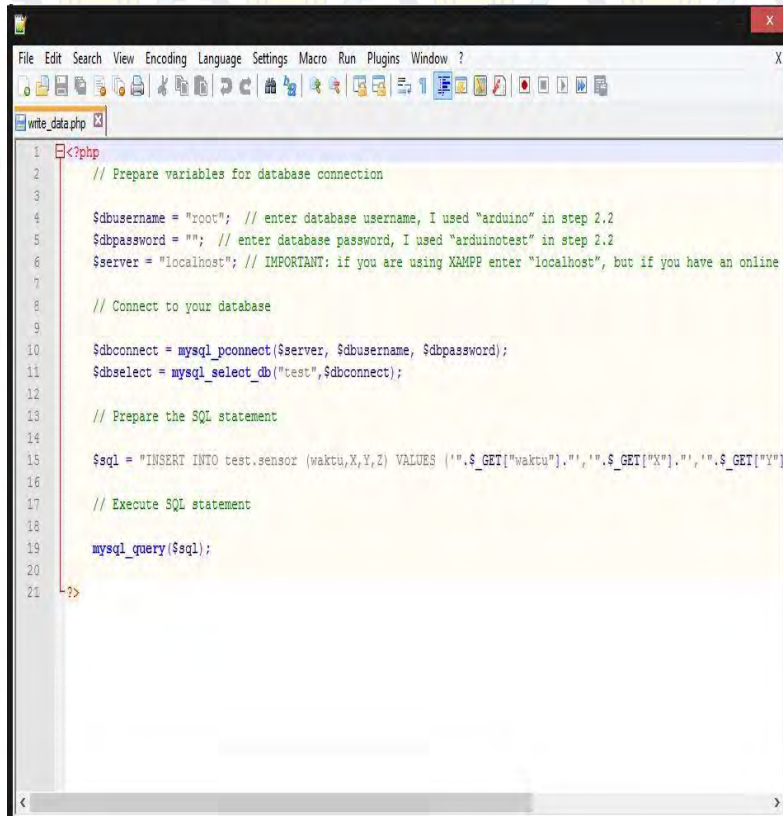
Gambar 3.15 Tampilan database MySQL

Setelah data serial masuk dan disimpan dalam database, selanjutnya adalah menampilkan data yang telah tersimpan dalam database tersebut pada halaman web.

Untuk menampilkan data tersebut pada halaman web harus menggunakan pemrograman web. Pada penelitian tugas akhir ini pemrograman web yang digunakan adalah pemrograman PHP.

3.4.2 PHP

PHP adalah singkatan dari "**PHP: Hypertext Preprocessor**", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web secara dinamis dan bisa digunakan bersamaan dengan HTML.

A screenshot of a text editor window titled 'write_data.php'. The window has a menu bar with 'File', 'Edit', 'Search', 'View', 'Encoding', 'Language', 'Settings', 'Macro', 'Run', 'Plugins', and 'Window'. Below the menu bar is a toolbar with various icons. The main text area contains PHP code for connecting to a MySQL database and inserting data. The code is as follows:

```
1 <?php
2 // Prepare variables for database connection
3
4 $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
5 $dbpassword = ""; // enter database password, I used "arduinotest" in step 2.2
6 $server = "localhost"; // IMPORTANT: if you are using XAMPP enter "localhost", but if you have an online
7
8 // Connect to your database
9
10 $dbconnect = mysql_pconnect($server, $dbusername, $dbpassword);
11 $dbselect = mysql_select_db("test", $dbconnect);
12
13 // Prepare the SQL statement
14
15 $sql = "INSERT INTO test.sensor (waktu,X,Y,Z) VALUES ('".$_GET["waktu"]."', '".$_GET["X"]."', '".$_GET["Y"]
16
17 // Execute SQL statement
18
19 mysql_query($sql);
20
21 ?>
```

Gambar 3.16 Tampilan PHP

3.5 Perancangan Jaringan Komunikasi *Wireless Local Area Network*

Pada perancangan jaringan komunikasi *Wireless Local Area Network (WLAN)* terdapat beberapa hal penting yang harus diperhatikan,

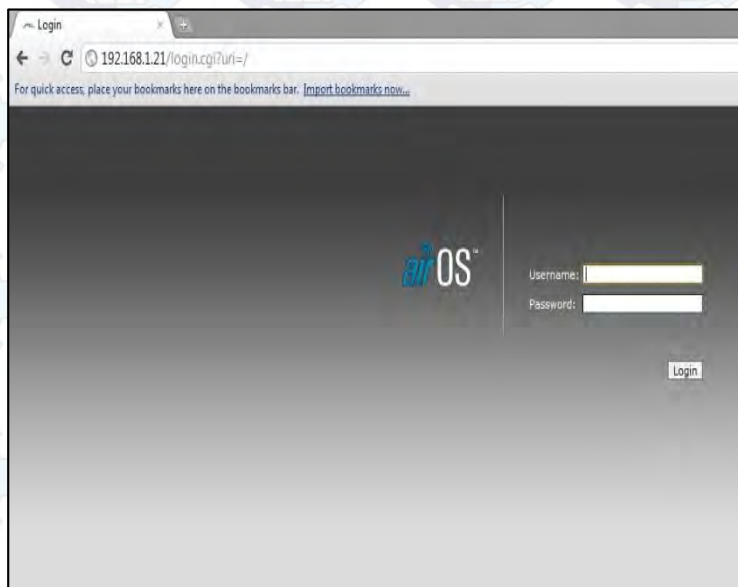
mulai dari perangkat keras, perangkat lunak, hingga cara kerja sistem secara keseluruhan.

Perangkat yang digunakan pada penelitian tugas akhir ini adalah perangkat *radiolink nanostationM2*. Perangkat *radiolink* ini dapat diatur sebagai *access point*, *bridge*, ataupun *router*.

Pada penelitian tugas akhir ini, *radiolink nanostationM2* digunakan sebagai *access point* untuk membuat jaringan komunikasi nirkabel pada sistem monitoring jembatan.

Untuk mengatur *nanostationM2* melakukan fungsi sebagai *access point*, ada beberapa langkah yang harus dilakukan:

1. Kabel power POE *nanostationM2* dihubungkan pada sumber listrik
2. Kabel UTP dari *nanostationM2* dihubungkan ke ethernet port pada PC
3. IP address PC diatur untuk satu kelas dengan default IP address dari *nanostationM2*
4. Buka browser kemudian buka default IP address *nanostationM2* pada browser



Gambar 3.17 Halaman *nanostation2*

5. Masuk dengan username dan password “ubnt”
6. Melakukan pengaturan dengan merubah fungsi radiolink *nanostationM2* menjadi access point dan atur beberapa fungsi yang diperlukan.

The screenshot shows the NanoStation2 web interface in a browser window. The address bar shows '192.168.1.21/index.cgi'. The interface has a top navigation bar with tabs: MAIN, WIRELESS, NETWORK, ADVANCED, SERVICES, and SYSTEM. The 'WIRELESS' tab is selected. The 'STATUS' section contains various configuration fields:

- Base Station SSID: SHM
- AirMax Quality: - %
- Frequency: 2412 MHz
- Antenna: Horizontal
- Security: none
- Uptime: 00:02:43
- LAN Cable: ON
- LAN MAC: 00:27:22:FF:36:BC
- WLAN MAC: 00:27:22:FE:36:BC
- Extra info: ---
- AirMax Capacity: - %
- Channel: 1
- Noise Floor: -96 dBm
- ACK Timeout: 162
- Date: 2012-04-23 10:01:36
- Host Name: UBNT
- LAN IP Address: 192.168.1.21
- WLAN IP Address: 192.168.1.21

Below the configuration fields are two tables for statistics:

LAN STATISTICS			
	Bytes	Packets	Errors
Received:	31506	174	0
Transmitted:	36422	102	0

WLAN STATISTICS			
	Bytes	Packets	Errors
Received:	0	0	0
Transmitted:	0	0	0

Gambar 3.18 Halaman Pengaturan nanostation m2

7. Melakukan uji komunikasi dengan melakukan ping pada IP address setiap perangkat, baik itu server maupun client.

Pada pengujian jaringan wireless local area network yang menggunakan perangkat nanostation m2 sebagai access point, dilakukan pengujian dengan monitoring jaringan untuk mendapatkan parameter jaringan seperti delay, paket loss.

Tabel 3.1 Pengukuran rata-rata daya terima access point

Jarak (m)	Daya Terima (dBm)
4	-70.24
8	-72.06
12	-71.92
16	-72.54
20	-76.81
24	-74.47
28	-85.33
32	-71.18
36	-68.59

Berdasarkan pengujian dengan daya pancar -10dBm didapatkan data bahwa daya terima yang didapatkan oleh *client* dari daya pancar *access point* tidak stabil. Hal itu disebabkan beberapa faktor seperti interferensi dari beberapa sinyal lain, losses, dan lainlain.

3.6 Performansi sistem

Perhitungan unjuk kerja dari sistem bertujuan untuk mengetahui kemampuan kerja dari sistem yang telah di bangun. Parameter unjuk kerja yang dianalisa adalah *packet loss*, *throughput* dan *delay* sistem.

Packet loss merupakan kegagalan transmisi pada suatu paket data untuk mencapai tujuannya. Kegagalan tersebut dapat disebabkan oleh beberapa kemungkinan seperti terjadinya *overload* trafik dalam jaringan, kerusakan pada media fisik, dan lain lain. Secara matematis *packet loss* dapat dituliskan dengan persamaan:

$$Packet Loss = \frac{Paket\ dikirim - Paket\ diterima}{Paket\ dikirim} \times 100\% \quad (3.1)$$

Troughput adalah jumlah data per unit waktu yang dikirimkan ke penerima tertentu dalam suatu jaringan, dari *node* jaringan, atau dari satu *node* ke *node* lain melalui *node* komunikasi. *Throughput* dapat dihitung dari hasil pembagian total paket data yang diterima dan durasi pengirimannya. Secara matematis dituliskan sebagai berikut.

$$\eta = \frac{N}{\tau} \times 8(bps) \quad (3.2)$$

Dengan :

η = Throughput

N = Paket data

τ = Total waktu untuk transmisi

Delay adalah selang waktu yang diperlukan oleh data yang dikirim dari *end device* hingga mencapai *coordinator*. Faktor-faktor yang mempengaruhi *end-to-end delay* adalah waktu untuk menemukan *route* pada jaringan, adanya *delay* pada eksekusi dari data yang masuk ke *router*, dan kondisi dari media tranmisi data. Secara matematis dapat dituliskan dengan persamaan berikut.

$$\Delta t = t_1 - t_0 \quad (3.3)$$

Dengan:

Δt = *end-to-end delay*

t_t = waktu terima

t_0 = waktu kirim

BAB 4

PENGUJIAN DAN ANALISA DATA

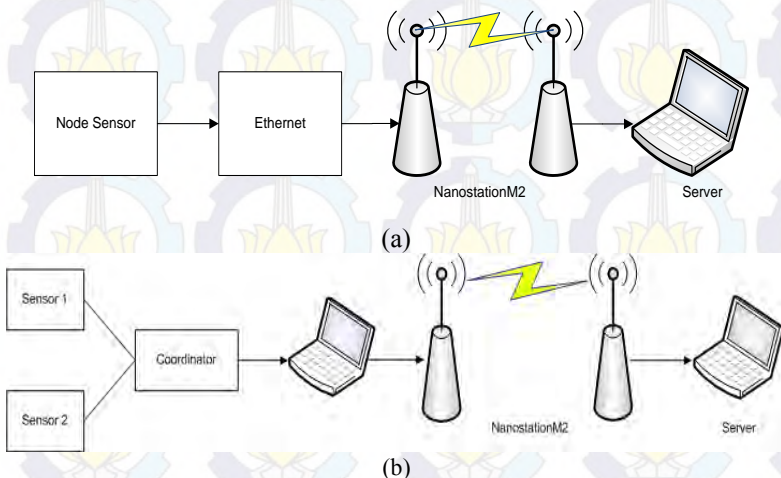
Setelah melakukan perancangan dan pengintegrasian perangkat, baik perangkat lunak maupun perangkat keras yang digunakan pada sistem, selanjutnya dilakukan pengujian pada sistem secara keseluruhan.

Pengujian dilakukan di jembatan antara gedung A dan B serta di halaman parkir dosen teknik elektro ITS. Pengujian ini dilakukan untuk mendapatkan data dan selanjutnya melakukan analisa unjuk kerja sistem secara keseluruhan.

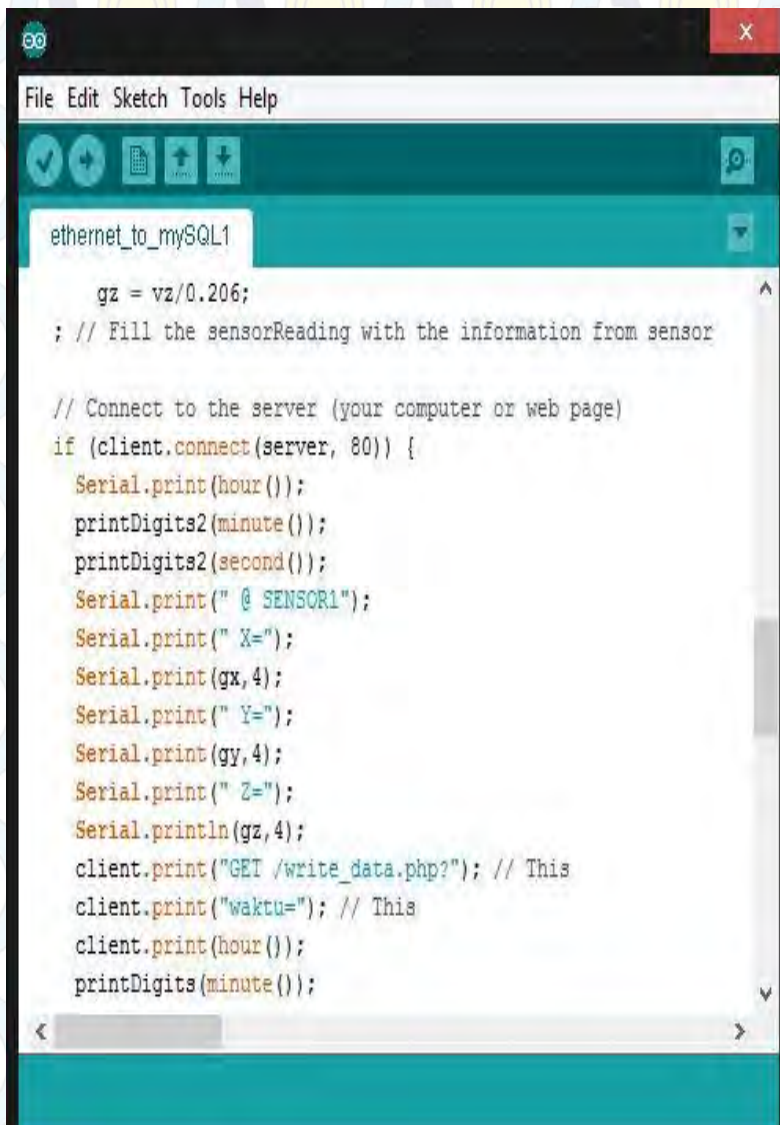
Pengujian sistem bertujuan untuk mendapatkan nilai dari parameter-parameter unjuk kerja jaringan pada sistem. Parameter yang dimaksud seperti yang telah dijelaskan pada bab sebelumnya yaitu packet loss, throughput, dan delay.

4.1 Pengujian node sensor (*end device*)

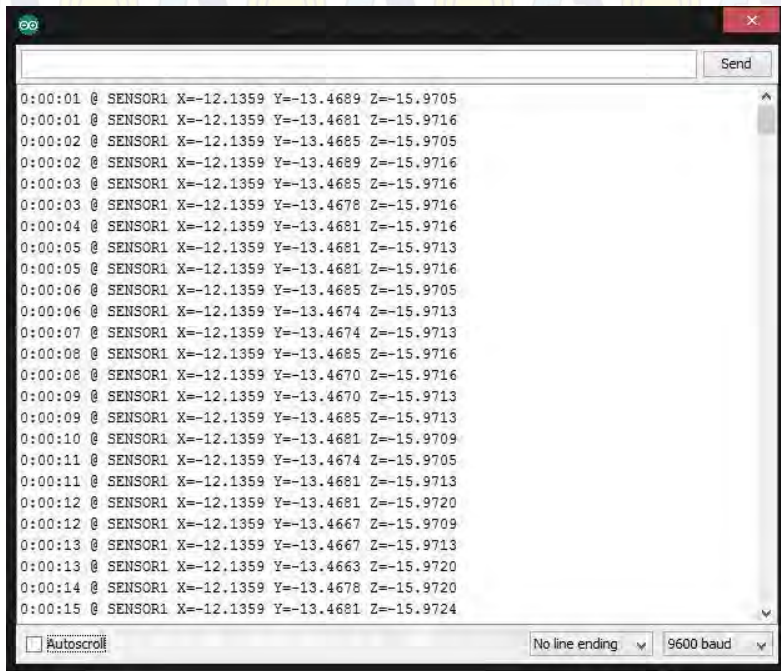
Pengujian dilakukan pada node sensor (*end device*). Pengujian dilakukan untuk mendapatkan data pembacaan sensor accelerometer.



Gambar 4.1 pengujian pada (a) skenario 1, (b) skenario 2



Gambar 4.2 Tampilan pemrograman arduino

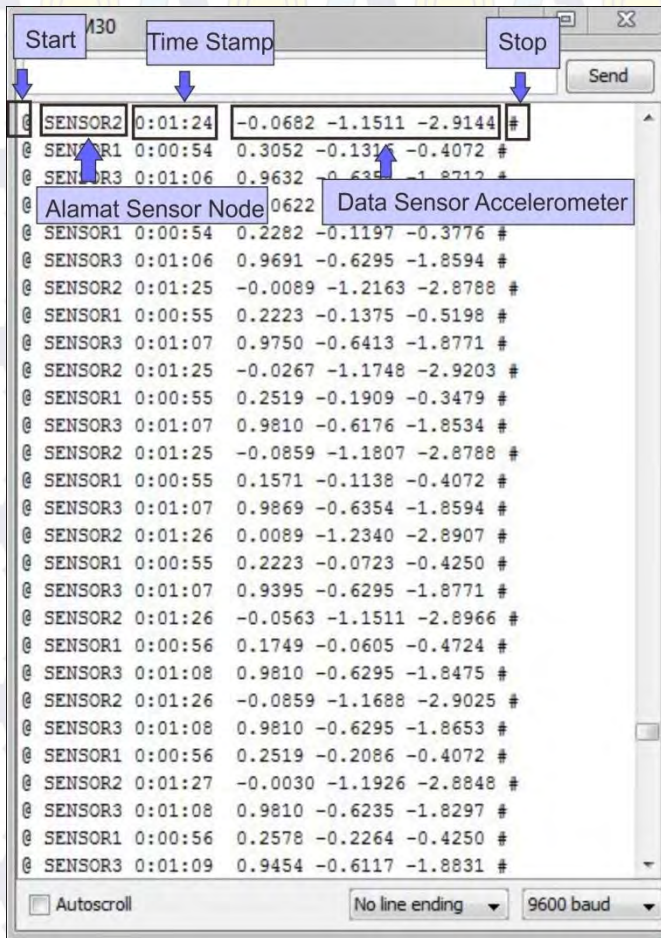


Gambar 4.3 window serial monitoring pada node sensor

Data pembacaan sensor accelerometer menunjukkan paket data seperti yang ada pada Gambar 4.3. Paket data terdiri dari start paket, nama node sensor, time stamp dan nilai pembacaan sensor dalam koordinat X, Y, Z serta stop paket.

4.2 Pengujian node koordinator

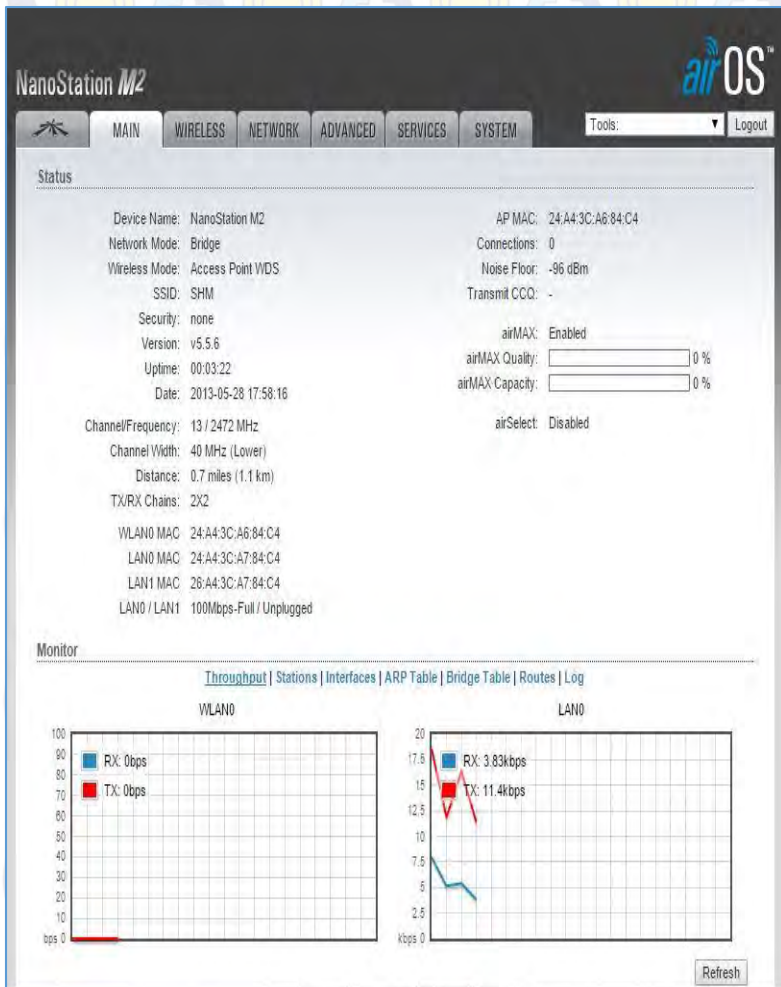
Pengujian dilakukan dengan melakukan komunikasi nirkabel antara node sensor dengan node koordinator. Node koordinator hanya berfungsi sebagai penerima paket data yang dikirimkan oleh dua node sensor.



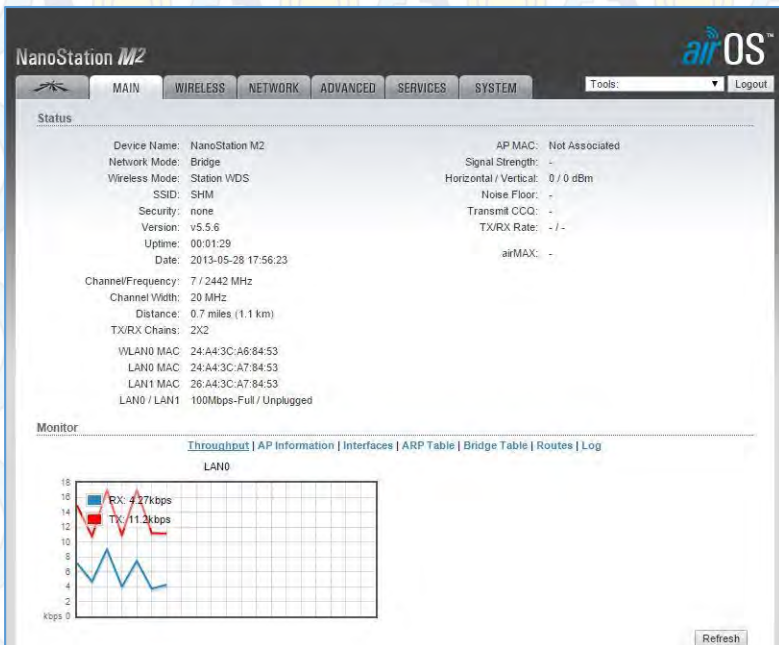
Gambar 4.4 window serial monitoring pada node coordinator

4.3 Pengujian NanoStation M2

Pengujian dilakukan dengan melakukan komunikasi WLAN point-to-point antara node sensor dengan server. Node sensor hanya berfungsi sebagai pengirim paket data yang akan diterima server. Dan nanostation akan di setting sebagai access point dan station.



(a)



(b)

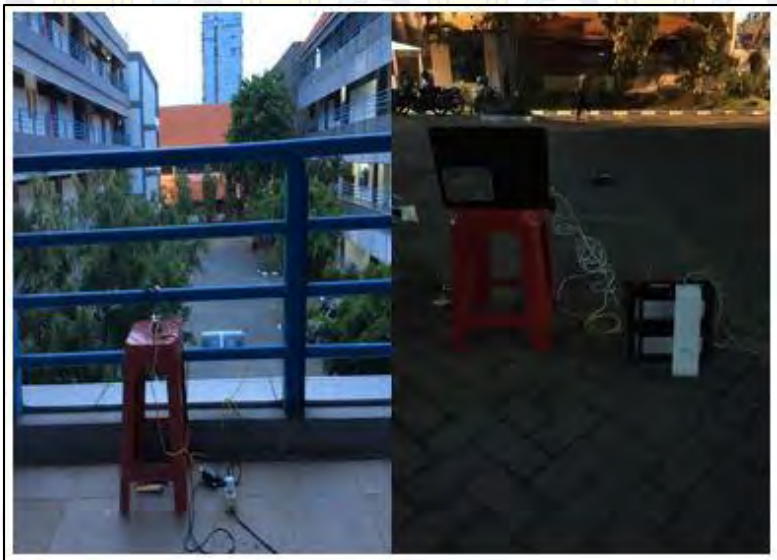
Gambar 4.5 nanostation sebagai access point (a), nanostation sebagai station (b)

4.4 Pengujian Komunikasi Node Sensor

Setelah melakukan pengujian pada node sensor seperti pada subbab 4.1 dan pengujian node koordinator seperti pada subbab 4.2 selanjutnya dilakukan pengukuran sebanyak lima kali dengan variasi jarak yang berbeda-beda pada skenario 1 dan skenario 2. Setiap pengukuran akan menunjukkan nilai parameter unjuk kerja dari node sensor.

4.4.1 Pengujian Node Sensor pada skenario ke- 1

Pada node sensor dilakukan pengujian untuk mendapatkan nilai throughput, packet loss dan delay.



Gambar 4.6 Pengujian komunikasi pada skenario ke-1

Parameter tersebut didapatkan dengan persamaan yang telah dijelaskan pada bab sebelumnya. Pengukuran dilakukan sebanyak lima kali dengan variasi jarak server terhadap node sensor yang berbeda yaitu $P1=40m$, $P2=30m$, $P3=25m$, $P4=15m$, dan $P5=10m$.

Tabel 4.1 Pengukuran node sensor pada skenario ke-1

Skenario ke-1	P1	P2	P3	P4	P5
Durasi Pengambilan Data (s)	300	300	300	300	300
Panjang Paket (byte)	99	99	99	99	99
Paket Terkirim (paket)	600	600	600	600	600
Paket Diterima (paket)	280	354	399	402	410
Throughput (bps)	92.4	116.82	131.67	132.66	135.3
Packet Loss (%)	53.33	41	33.5	33	31.6

- Perhitungan pengukuran pertama :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 280)}{300} = 92.4 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 280)}{600} \times 100\% = 53.33\%$$

- Perhitungan pengukuran kedua :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 354)}{300} = 116.82 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 354)}{600} \times 100\% = 41\%$$

- Perhitungan pengukuran ketiga :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 399)}{300} = 131.67 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 399)}{600} \times 100\% = 33.5\%$$

- Perhitungan pengukuran keempat :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 402)}{300} = 132.66 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 402)}{600} \times 100\% = 33\%$$

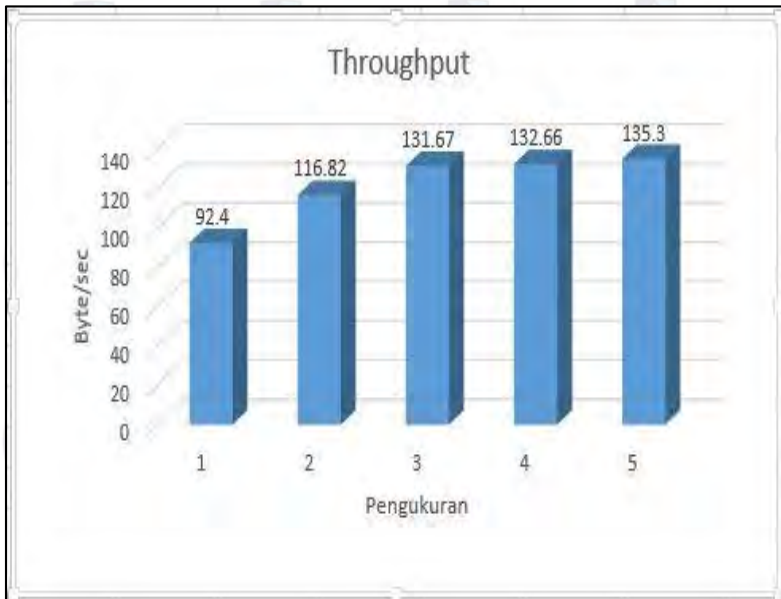
- Perhitungan pengukuran kelima :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

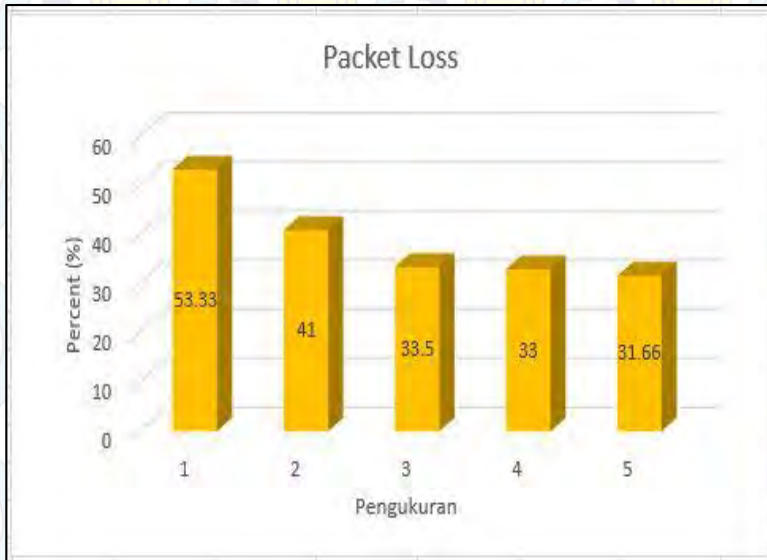
$$\text{Throughput} = \frac{(99 \times 410)}{300} = 135.3 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 410)}{600} \times 100\% = 31.66\%$$



Gambar 4.7 Grafik pengukuran throughput node sensor



Gambar 4.8 Grafik pengukuran packet loss node sensor

Tabel 4.2 Pengukuran rata-rata throughput dan packet loss node sensor

Pengukuran	Throughput	Packet Loss
1	92.4	53.33
2	116.82	41
3	131.67	33.5
4	132.66	33
5	135.3	31.66
rata-rata	121.77	38.486

Melalui proses pengujian ini didapatkan data throughput dan packet loss node sensor1 seperti pada **Tabel 4.2** dimana banyak paket yang hilang akan berpengaruh pada nilai throughput. Hal tersebut sesuai dengan perhitungan teori.

Perbedaan nilai packet loss dan throughput pada tiap pengujian disebabkan oleh beberapa faktor seperti jarak server, proses komputasi

mikrokontroler, dan adanya noise atau gangguan saat proses komunikasi data berlangsung.

4.4.2 Pengujian Node Sensor pada skenario ke-2

Pada pengujian node sensor skenario ke-2 dilakukan pengujian pada sensor1 dan sensor2 untuk mendapatkan nilai throughput, packet loss dan delay.



Gambar 4.9 Pengujian komunikasi pada skenario ke-2

Parameter tersebut didapatkan dengan persamaan yang telah dijelaskan pada bab sebelumnya. Pengukuran dilakukan sebanyak lima kali dengan variasi jarak server terhadap node sensor yang berbeda yaitu $P1=40m$, $P2=30m$, $P3=25m$, $P4=15m$, dan $P5=10m$.

4.4.2.1 Node Sensor 1

Pengujian komunikasi pada node sensor 1 dilakukan dengan mengirimkan data pembacaan nilai sensor accelerometer menuju node coordinator.

Tabel 4.3 Pengukuran node sensor 1 pada skenario ke-2

Skenario ke-2	Sensor 1				
	P1	P2	P3	P4	P5
Durasi Pengambilan Data (s)	300	300	300	300	300
Panjang Paket (byte)	99	99	99	99	99
Paket Terkirim (paket)	600	600	600	600	600
Paket Diterima (paket)	113	129	192	147	147
Throughput (bps)	37.29	42.57	63.36	48.51	48.51
Packet Loss (%)	81.16	78.5	68	75.5	75.5

- Perhitungan pengukuran pertama :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 113)}{300} = 37.29 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 113)}{600} \times 100\% = 81.16\%$$

- Perhitungan pengukuran kedua :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 129)}{300} = 42.57 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 129)}{600} \times 100\% = 78.5\%$$

- Perhitungan pengukuran ketiga :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 192)}{300} = 63.36 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 192)}{600} \times 100\% = 68\%$$

- Perhitungan pengukuran keempat :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 147)}{300} = 48.51 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 147)}{600} \times 100\% = 75.5\%$$

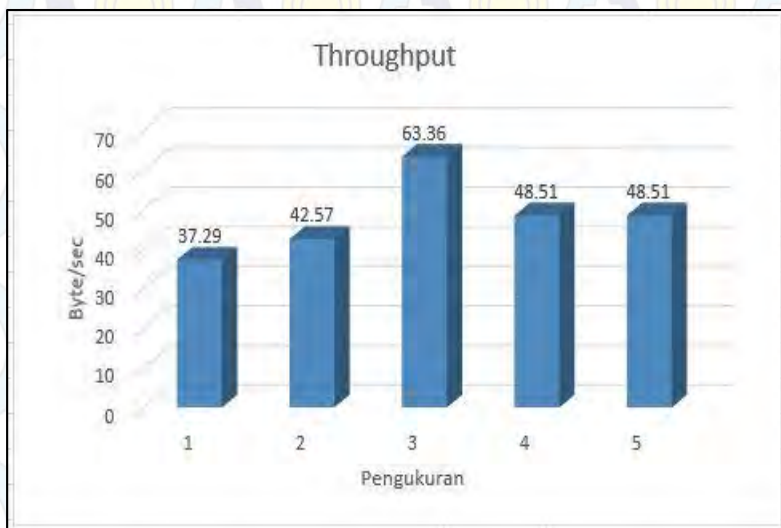
- Perhitungan pengukuran kelima :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

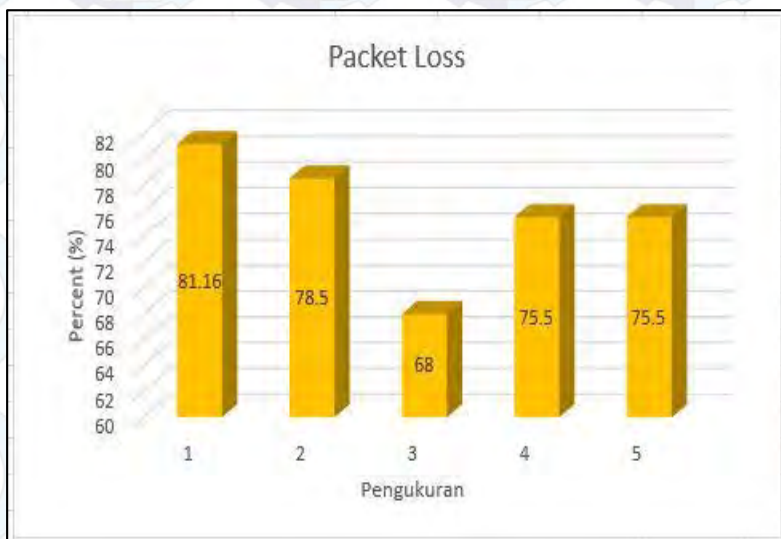
$$\text{Throughput} = \frac{(99 \times 147)}{300} = 48.51 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 147)}{600} \times 100\% = 75.5\%$$



Gambar 4.10 Grafik pengukuran throughput node sensor 1



Gambar 4.11 Grafik pengukuran packet loss node sensor 1

Tabel 4.4 Pengukuran rata-rata throughput dan packet loss node sensor1

Pengukuran	Throughput	Packet Loss
1	37.29	81.16
2	42.57	78.5
3	63.36	68
4	48.51	75.5
5	48.51	75.5
rata-rata	48.048	75.732

Melalui proses pengujian ini didapatkan data throughput dan packet loss node sensor1 seperti pada **Tabel 4.4** dimana banyak paket yang hilang akan berpengaruh pada nilai throughput. Hal tersebut sesuai dengan perhitungan teori.

Perbedaan nilai packet loss dan throughput pada tiap pengujian disebabkan oleh beberapa faktor seperti proses komputasi mikrokontroler dan adanya noise atau gangguan saat proses komunikasi data berlangsung.

4.4.2.2 Node Sensor 2

Pengujian komunikasi pada node sensor 2 dilakukan dengan mengirimkan data pembacaan nilai sensor accelerometer menuju node coordinator.

Tabel 4.5 Pengukuran node sensor 2 pada skenario ke-2

Skenario ke-2	Sensor 2				
	P1	P2	P3	P4	P5
Durasi Pengambilan Data (s)	300	300	300	300	300
Panjang Paket (byte)	99	99	99	99	99
Paket Terkirim (paket)	600	600	600	600	600
Paket Diterima (paket)	107	140	84	130	138
Throughput (bps)	35.31	46.2	27.72	42.9	45.54
Packet Loss (%)	82.16	76.66	86	78.33	77

- Perhitungan pengukuran pertama :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 107)}{300} = 35.31 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 107)}{600} \times 100\% = 82.16\%$$

- Perhitungan pengukuran kedua :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 140)}{300} = 46.2 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 140)}{600} \times 100\% = 76.66\%$$

- Perhitungan pengukuran ketiga :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 84)}{300} = 27.72 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 84)}{600} \times 100\% = 86\%$$

- Perhitungan pengukuran keempat :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 130)}{300} = 42.9 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 130)}{600} \times 100\% = 78.33\%$$

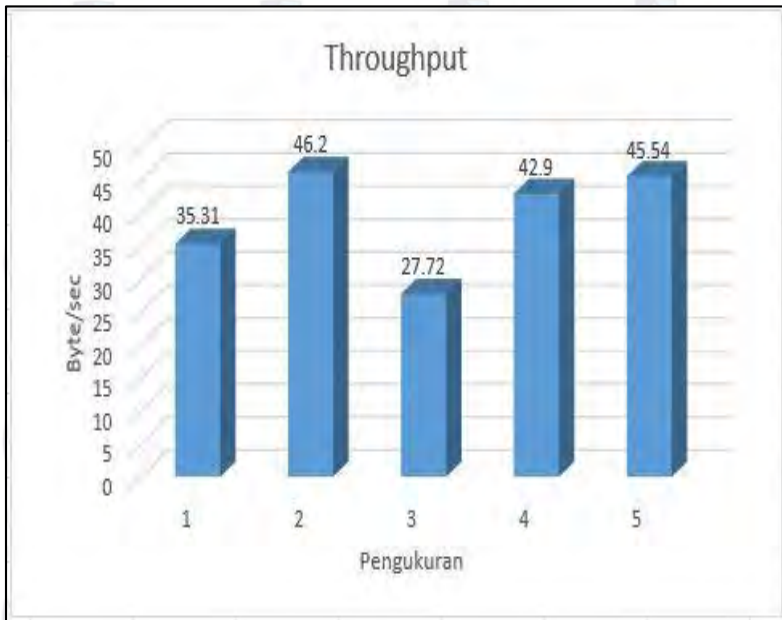
- Perhitungan pengukuran kelima :

$$\text{Throughput} = \frac{(\text{Panjang per paket} \times \text{Jumlah paket terima})}{\text{Durasi pengambilan data}}$$

$$\text{Throughput} = \frac{(99 \times 138)}{300} = 45.54 \text{ Bps}$$

$$\text{Packet Loss} = \frac{(\text{Paket terkirim} - \text{Paket terima})}{\text{Paket terkirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{(600 - 138)}{600} \times 100\% = 77\%$$



Gambar 4.12 Grafik pengukuran throughput node sensor 2



Gambar 4.13 Grafik pengukuran packet loss node sensor 2

Tabel 4.6 Pengukuran rata-rata throughput dan packet loss node sensor2

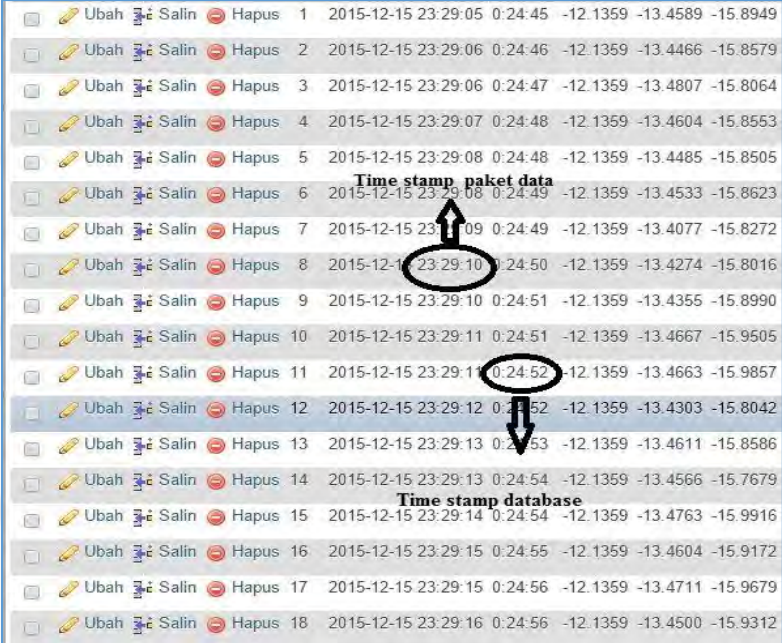
Pengukuran	Throughput	Packet Loss
1	35.31	82.16
2	46.2	76.66
3	27.72	86
4	42.9	78.33
5	45.54	77
rata-rata	39.534	80.03

Melalui proses pengujian ini didapatkan data throughput dan packet loss node sensor2 seperti pada **Tabel 4.6** dimana banyak paket yang hilang akan berpengaruh pada nilai throughput. Hal tersebut sesuai dengan perhitungan teori.

Perbedaan nilai packet loss dan throughput pada tiap pengujian disebabkan oleh beberapa faktor seperti proses komputasi mikrokontroler dan adanya noise atau gangguan saat proses komunikasi data berlangsung.

4.5 Pengujian Delay Node Sensor

Pengujian delay dilakukan dengan mengirimkan data pembacaan sensor accelerometer pada node sensor menuju database. Database yang digunakan sebagai penyimpanan data adalah perangkat lunak pemrograman MySQL.



<input type="checkbox"/>	Ubah Salin Hapus	1	2015-12-15 23:29:05	0.24:45	-12.1359	-13.4589	-15.8949
<input type="checkbox"/>	Ubah Salin Hapus	2	2015-12-15 23:29:06	0.24:46	-12.1359	-13.4466	-15.8579
<input type="checkbox"/>	Ubah Salin Hapus	3	2015-12-15 23:29:06	0.24:47	-12.1359	-13.4807	-15.8064
<input type="checkbox"/>	Ubah Salin Hapus	4	2015-12-15 23:29:07	0.24:48	-12.1359	-13.4604	-15.8553
<input type="checkbox"/>	Ubah Salin Hapus	5	2015-12-15 23:29:08	0.24:48	-12.1359	-13.4485	-15.8505
<input type="checkbox"/>	Ubah Salin Hapus	6	2015-12-15 23:29:08	0.24:49	-12.1359	-13.4533	-15.8623
<input type="checkbox"/>	Ubah Salin Hapus	7	2015-12-15 23:29:09	0.24:49	-12.1359	-13.4077	-15.8272
<input type="checkbox"/>	Ubah Salin Hapus	8	2015-12-15 23:29:10	0.24:50	-12.1359	-13.4274	-15.8016
<input type="checkbox"/>	Ubah Salin Hapus	9	2015-12-15 23:29:10	0.24:51	-12.1359	-13.4355	-15.8990
<input type="checkbox"/>	Ubah Salin Hapus	10	2015-12-15 23:29:11	0.24:51	-12.1359	-13.4667	-15.9505
<input type="checkbox"/>	Ubah Salin Hapus	11	2015-12-15 23:29:11	0.24:52	-12.1359	-13.4663	-15.9857
<input type="checkbox"/>	Ubah Salin Hapus	12	2015-12-15 23:29:12	0.24:52	-12.1359	-13.4303	-15.8042
<input type="checkbox"/>	Ubah Salin Hapus	13	2015-12-15 23:29:13	0.24:53	-12.1359	-13.4611	-15.8586
<input type="checkbox"/>	Ubah Salin Hapus	14	2015-12-15 23:29:13	0.24:54	-12.1359	-13.4566	-15.7679
<input type="checkbox"/>	Ubah Salin Hapus	15	2015-12-15 23:29:14	0.24:54	-12.1359	-13.4763	-15.9916
<input type="checkbox"/>	Ubah Salin Hapus	16	2015-12-15 23:29:15	0.24:55	-12.1359	-13.4604	-15.9172
<input type="checkbox"/>	Ubah Salin Hapus	17	2015-12-15 23:29:15	0.24:56	-12.1359	-13.4711	-15.9679
<input type="checkbox"/>	Ubah Salin Hapus	18	2015-12-15 23:29:16	0.24:56	-12.1359	-13.4500	-15.9312

Gambar 4.14 Tampilan database MySQL

Paket data yang dikirim node sensor menuju ke database memiliki data waktu pengiriman. Data waktu pengiriman tersebut akan dibandingkan dengan data time stamp yang telah dibuat pada pemrograman MySQL. Selisih waktu tersebut menjadi parameter dalam perhitungan delay.

Dengan pengambilan data sebanyak 100 paket dari node sensor tiap pengukuran maka didapatkan delay end to end tiap pengukuran seperti pada **Tabel 4.7**:

Tabel 4.7 Pengukuran rata-rata delay sensor tiap pengukuran

Skenario ke- 1	
Node Sensor	Rata-Rata Delay
Sensor	1.26 detik
Skenario ke- 2	
Node Sensor	Rata-Rata Delay
Sensor 1	1.2 detik
Sensor 2	1.02 detik

4.6 Analisa Sistem

Nilai unjuk kerja jaringan dari sistem *wireless sensor network* pada proses pengujian pada jembatan AJ Teknik Elektro ITS yaitu *packet loss*, *throughput* dan *delay* dipengaruhi oleh beberapa faktor diantaranya komputasi pada mikrokontroler arduino, jarak komunikasi, serta kondisi lingkungan sekitar pada saat proses pengujian.

Delay end to end pada node sensor dari hasil pengujian yang didapatkan lebih dari satu detik lebih disebabkan oleh waktu transmisi serta proses komputasi mikrokontroler.

Komputasi pada mikrokontroler mempengaruhi proses pengiriman dan penerimaan data. Pada setiap node baik node sensor maupun node koordinator memiliki kerja ganda. Pada node sensor mikrokontroler berfungsi untuk mengolah data pembacaan sensor accelerometer menjadi paket data dan melakukan proses pengiriman paket data menuju database.

Besarnya nilai *throughput* berdasarkan pengujian diketahui berbanding terbalik dengan besarnya paket data yang hilang saat proses pengiriman data. Jika paket yang dikirim banyak yang hilang, maka nilai dari *throughput* akan rendah.

Packet Loss atau paket yang hilang pada saat proses komunikasi data antara node sensor menuju database disebabkan oleh beberapa faktor seperti antrian paket data pada node koordinator dan kesalahan pembacaan pada mikrokontroler yang menyebabkan paket error.



3	2016-01-12 23:01:53	@ SENSOR1	1.18.17	-7.6550	-6.3196	-12.7298	#
4	2016-01-12 23:01:53	@ SENSOR2	0:17.17	5.2130	5.4103	-2.0668	#
5	2016-01-12 23:01:54	@ SENSOR1	1.18.18	-3.9505	-2.6981	-10.0922	#
6	2016-01-12 23:01:57	@ SENSOR2	0:17.18	5.1003	5.3	@ SENSOR2 0:17.21	2....
7	2016-01-12 23:01:57	@ SENSOR1	1.18.22	3.2866	4.4501	-4.6748	#
8	2016-01-12 23:01:58	@ SENSOR2	0:17.22	2.5220	2.8498	-4.0346	#
9	2016-01-12 23:01:58	@ SENSOR1	1.18.22	3.5059	4.6813	-4.4733	#
10	2016-01-12 23:01:58	@ SENSOR2	0:17.22	2.2672	2.6127	-4.2302	#
11	2016-01-12 23:01:59	@ SENSOR1	1.18.23	3.6600	4.9006	-4.2954	#
12	2016-01-12 23:01:59	@ SENSOR2	0:17.23	1.8700	2.2334	-4.5207	#
13	2016-01-12 23:02:07	@ SENSOR1	1.18.23	3.6719	4.9480	-4.2159M=IA 1:18...	
14	2016-01-12 23:02:07	@ SENSOR2	0:17.31	4.2646	4.0471	-3.2108	#
15	2016-01-12 23:02:08	@ SENSOR1	1.18.32	2.0478	3.2706	-5.3208	#
16	2016-01-12 23:02:08	@ SENSOR2	0:17.32	5.2782	5.2207	-2.3454	#
17	2016-01-12 23:02:08	@ SENSOR1	1.18.33	1.6270	2.8498	-5.6409	#
18	2016-01-12 23:02:09	@ SENSOR2	0:17.33	5.5271	5.7008	-1.9779	#
19	2016-01-12 23:02:12	@ SENSOR1	1.18.33	1.1943	2.4171	-5.96@ SENSOR2 0...	
20	2016-01-12 23:02:13	@ SENSOR1	1.18.37	-1.5085	-0.2620	-8.2548	#

Gambar 4.15 Data error pada database

Berdasarkan pengujian tersebut sistem *wireless sensor network* dapat diimplementasikan pada Jembatan bentang panjang Surabaya-Madura dengan beberapa perangkat penyusun tambahan yang mendukung implementasi *wireless sensor network* pada kondisi nyata di Jembatan bentang panjang Surabaya-Madura.

Skenario komunikasi data *wireless sensor network* untuk implementasi pada jembatan bentang panjang sama dengan pada saat pengujian dengan node sensor mengirimkan paket data pembacaan sensor menuju ke node koordinator. Node koordinator terhubung dengan *field server* yang tersambung dengan jaringan lokal melalui komunikasi *radio-link* menggunakan nanostation.

Penempatan posisi node sensor dengan node koordinator sama dengan pada proses pengujian dengan satu node koordinator berada diantara dua node sensor. Hal ini dikarenakan untuk mendapatkan nilai unjuk kerja jaringan yang optimal.

BAB 5

KESIMPULAN

Rangkaian penelitian yang telah dilakukan akan memberikan kesimpulan. Dari pembahasan pada bab empat dan masalah-masalah yang terjadi pengerjaan tugas akhir ini akan menjadi saran untuk dijadikan pertimbangan dalam melakukan penelitian selanjutnya mengenai sistem komunikasi *wireless sensor network* di kemudian hari.

5.1 Kesimpulan

Berdasarkan hasil penelitian mengenai sistem komunikasi data untuk wsn modul pada jembatan bentang panjang surabaya-madura dan juga data yang di dapat dari hasil pengukuran kinerja sistem yang dibangun dapat disimpulkan :

1. Pada penelitian ini menunjukkan nilai unjuk kerja rata-rata dari sistem *wireless sensor network* sebagai berikut :
 - Skenario ke-1 :
 - a. Throughput node sensor adalah 121.77 Bps
 - b. Packet Loss node sensor adalah 38.486 %
 - c. Delay node sensor adalah 1.26 detik
 - Skenario ke-2 :
 - a. Throughput node sensor 1 adalah 48.048 Bps
Throughput node sensor 2 adalah 39.534 Bps
 - b. Packet Loss node sensor 1 adalah 75.732 %
Packet Loss node sensor 2 adalah 80.03 %
 - c. Delay node sensor 1 adalah 1.2 detik
Delay node sensor 2 adalah 1.02 detik
2. Jarak posisi antar node dan kondisi lingkungan sekitar sistem sangat berpengaruh terhadap unjuk kerja sistem komunikasi data *wireless sensor network* secara keseluruhan.
3. Modul *wireless sensor network* sangat sensitif terhadap gangguan saat proses komunikasi. Untuk memperoleh nilai unjuk kerja yang maksimal diperlukan kondisi komunikasi yang Line of Sight dengan gangguan seminimal mungkin.

5.2 Saran

Berdasarkan hasil penelitian tugas akhir ini, ada beberapa poin yang dapat dijadikan pertimbangan dan saran untuk pengembangan penelitian *wireless sensor network* dikemudian hari:

1. Penggunaan PC atau Laptop sebagai *field server* dapat diganti dengan perangkat yang memiliki fungsi yang sama seperti moxa. Perangkat tersebut lebih dapat digunakan dilapangan.
2. Pada pengembangan penelitian selanjutnya, pada sisi node koordinator dapat ditambahkan beberapa algoritma untuk penjadwalan dalam penerimaan paket data dari node koordinator untuk mengurangi delay antrian pada saat penerimaan paket data.

LAMPIRAN

Listing Program Mikrokontroler Arduino sebagai Node Sensor

```
#include <SPI.h>
#include <Ethernet.h>
#include <Time.h>
#define TIME_HEADER "T"
#define TIME_REQUEST 7
int x2pin = A0;
int y2pin = A1;
int z2pin = A2;
float x2,y2,z2;
float vx, vy, vz, gx, gy, gz;
//byte pesan[1];
//byte b[30];
//byte ACK[2];
int i;
int x,y;
#define ROUTER_R "R"
#define ROUTER_P "P"

byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0D, 0xC4, 0x99 };

// Enter the IP address for Arduino, as mentioned we will use
192.168.0.16
// Be careful to use , instead of . when you enter the address here
IPAddress ip(192, 168, 1, 1);

char server[] = "192.168.1.4"; // IMPORTANT: If you are using
XAMPP you will have to find out the IP address of your computer and
put it here (it is explained in previous article). If you have a web page,
enter its address (ie. "www.yourwebpage.com")
```



```
// Initialize the Ethernet server library
EthernetClient client;
```

```
void setup() {
```

```
    // Serial.begin starts the serial connection between computer and
    Arduino
```

```
    Serial.begin(9600);
```

```
    // start the Ethernet connection
```

```
    Ethernet.begin(mac, ip);
```

```
}
```

```
void loop() {
```

```
    x2=analogRead(x2pin);
```

```
    y2=analogRead(y2pin);
```

```
    z2=analogRead(z2pin);
```

```
    vx = (x2*5/65520)-2.50;
```

```
    gx = vx/0.206;
```

```
    vy = (y2*5/65520)-2.8;
```

```
    gy = vy/0.206;
```

```
    vz = (z2*5/65520)-3.3;
```

```
    gz = vz/0.206;
```

```
; // Fill the sensorReading with the information from sensor
```

```
// Connect to the server (your computer or web page)
```

```
if (client.connect(server, 80)) {
```

```
    Serial.print(hour());
```

```
    printDigits2(minute());
```

```
    printDigits2(second());
```

```
    Serial.print("&X=");
```

```
    Serial.print(gx,4);
```

```
    Serial.print("&Y=");
```

```
    Serial.print(gy,4);
```

```
    Serial.print("&Z=");
```

```
    Serial.print(gz,4);
```

```
    client.print("GET /write_data.php?"); // This
```

```
    client.print("waktu="); // This
```

```

client.print(hour());
printDigits(minute());
printDigits(second());
client.print("&X=");
client.print(gx,4);
client.print("&Y=");
client.print(gy,4);
client.print("&Z=");
client.print(gz,4);
//client.print(" ");
    // And this is what we did in the testing section above. We are
    making a GET request just like we would from our browser but now
    with live data from the sensor
    client.println(" HTTP/1.1"); // Part of the GET request
    client.println("Host: 192.168.1.4"); // IMPORTANT: If you are using
    XAMPP you will have to find out the IP address of your computer and
    put it here (it is explained in previous article). If you have a web page,
    enter its address (ie.Host: "www.yourwebpage.com")
    client.println("Connection: close"); // Part of the GET request telling
    the server that we are over transmitting the message
    client.println(); // Empty line
    client.println(); // Empty line
    client.stop(); // Closing connection to server
}

else {
    // If Arduino can't connect to the server (your computer or web page)
    Serial.println("--> connection failed/n");
}

    // Give the server some time to receive the data and store it. I used 10
    seconds here. Be advised when delaying. If u use a short delay, the
    server might not capture data because of Arduino transmitting new data
    too soon.
    delay(500);
}

void printDigits(int digits){

```



```

client.print(":");
if(digits < 10)
  client.print('0');
client.print(digits);
}
void printDigits2(int digits){
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

```

Listing Program Mikrokontroler Arduino sebagai Node Coordinator

```

/*
XBeeEcho
Reply with whatever you receive over the serial port
*/
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  while (Serial.available() ) {
    Serial.write(Serial.read()); // reply with whatever you receive
  }
}

```

Listing Program Python

```

#!/usr/bin/python
import serial
import MySQLdb
import time
import datetime
import re
from threading import Thread
import subprocess

```



```
dbConn = MySQLdb.connect("127.0.0.1","root","", "dataterima") or die  
("could not connect to database")
```

```
ser = serial.Serial('COM7', 9600)
```

```
data = ser.readline()
```

```
while data:
```

```
    stream = ser.readline()
```

```
    cursor = dbConn.cursor()
```

```
    try:
```

```
        cursor.execute("INSERT INTO tabel (data) VALUES
```

```
(%s)", [stream])
```

```
        dbConn.commit() #commit the insert
```

```
    except MySQLdb.Error as e:
```

```
        print e
```

```
        continue
```

```
    finally:
```

```
        cursor.close()
```

Listing Program PHP

```
<?php
```

```
    // Prepare variables for database connection
```

```
    $dbusername = "root"; // enter database username, I used "arduino"  
in step 2.2
```

```
    $dbpassword = ""; // enter database password, I used "arduinotest" in  
step 2.2
```

```
    $server = "localhost"; // IMPORTANT: if you are using XAMPP  
enter "localhost", but if you have an online website enter its address,  
ie."www.yourwebsite.com"
```

```
    // Connect to your database
```

```
    $dbconnect = mysql_pconnect($server, $dbusername, $dbpassword);
```

```
    $dbselect = mysql_select_db("test",$dbconnect);
```

```
    // Prepare the SQL statement
```

```
$sql = "INSERT INTO test.sensor (waktu,X,Y,Z) VALUES  
('".$_GET["waktu"]."',".$_GET["X"]."',".$_GET["Y"]."',".$_GET["Z"]  
.'");";
```

```
// Execute SQL statement
```

```
mysql_query($sql);
```

```
?>
```

DAFTAR PUSTAKA

- [1] Bambang Suhendro, "SHMS Jembatan Suramadu Sebagai Penunjang Preservasi dan Pengembangan Teknologi Jembatan", Jurusan Teknik Sipil dan Lingkungan, Fakultas Teknik, Universitas Gadjah Mada, 2010.
- [2] E. G. Straser and A. S. Kiremidjian, "*A modular, wireless damage monitoring system for structures,*" John A. Blume Earthquake Eng. Center, Dept. Civil Environ. Eng., Stanford Univ., Stanford, CA, Tech. Rep. 128, 1998.
- [3] A. S. Kiremidjian, E. G. Straser, T. H. Meng, K. Law, and H. Soon, "*Structural damage monitoring for civil structures,*" in Proc. Int. Workshop Struct. Health Monit., Stanford, CA, 1997, pp. 371–382.
- [4] Wei Chen, Miguel R. D. Rodrigues and Ian J. Wassell, "*A Frechet Mean Approach for Compressive Sensing Data Acquisition and Reconstruction in Wireless Sensor Networks*" IEEE Transactions on wireless communications, vol. 11, no. 10, October 2012.
- [5] Margolis Michael (2012). *Arduino Cookbook 2nd Edition*. : United State of America : O'Reilly Media Inc.
- [6] Sohraby Kazem, Minoli Daniel, and Znati Taieb (2007). *Wireless Sensor Networks Technology, Protocols, and Applications*. Canada : Wiley-Interscience.
- [7] Eko Setijadi, Suwadi, Slamet BP, Priyo S, Faimun, Arie Febry, Muntaqo A, In'am, Evy Nur A, "*Design of Large Scale Structural Health Monitoring System for Long-Span Bridges Based on Wireless Sensor Network*", International Joint Conference on Awareness Science and Technology & Ubi-Media Computing 2013.
- [8] Hebel Martin, Bricker George (2010). *Getting Started with Xbee RF Modules*. USA : Parallax Inc
- [9] www.arduino.cc
- [10] www.digi.com

BIOGRAFI PENULIS



Arvid Prasetya Cahya Lahir di Jakarta 28 November 1992. Penulis merupakan anak kedua dari pasangan ayah Mirzah Heru Cahya dan ibu Sri Rahayu Setyowati. Penulis memulai pendidikannya di TK Aisiyah Pamulang pada tahun 1997, kemudian melanjutkan pendidikan di SD Muhammadiyah 12 Pamulang pada tahun 1998. Tahun 2004 penulis melanjutkan pendidikannya di SMP Muhammadiyah 22 lalu melanjutkan pendidikannya di SMAN 1 Kota Tangerang Selatan pada tahun 2007. Dan pada tahun 2010 penulis meneruskan studi ke jenjang perkuliahan di Jurusan Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya